



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE141599

**OPTIMASI PEMBEBANAN GENERATOR UNTUK
MEMINIMALKAN BIAYA OPERASI PEMBANGKITAN
MENGUNAKAN *DYNAMIC FORMULATION*
*TECHNIQUE***

**Oktarina Ratri Wijayanti
NRP 2213 106 001**

**Dosen Pembimbing
Prof. Ir. Ontoseno Penangsang, M. Sc, Ph. D.
Dr. Rony Seto Wibowo, ST., MT.**

**JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016**



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 141599

***GENERATOR LOAD OPTIMIZATION FOR
MINIMALIZING GENERATION COST USING DYNAMIC
FORMULATION TECHNIQUE***

**Oktarina Ratri Wijayanti
NRP 2213 106 001**

Advisor

**Prof. Ir. Ontoseno Penangsang, M. Sc, Ph. D.
Dr. Rony Seto Wibowo, ST., MT.**

**ELECTRICAL ENGINEERING DEPARTEMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

**OPTIMASI PEMBEBANAN GENERATOR UNTUK
MEMINIMALKAN BIAYA OPERASI PEMBANGKITAN
MENGUNAKAN *DYNAMIC FORMULATION TECHNIQUE***

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

Pada

**Bidang Studi Teknik Sistem Tenaga
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I,

Dosen Pembimbing II,

Prof. Ir. Ontoseno Penangsang, M. Sc, Ph. D.
NIP. 1949 07 15 1974 12 1001

Dr. Rony Seto Wibowo, ST., MT.
NIP. 1974 11 29 2000 12 1001



**SURABAYA
JANUARI 2016**

ABSTRAK

“Optimasi Pembebanan Generator Untuk Meminimalkan Biaya Operasi Pembangkitan Menggunakan *Dynamic Formulation Technique*”

Oktarina Ratri Wijayanti
2213106001

Dosen pembimbing 1 : Prof. Ir. Ontoseno Penangsang, M.Sc, Ph.D
Dosen Pembimbing 2 : Dr. Rony Seto Wibowo, ST., MT.

Abstrak:

Economic Dispatch (ED) merupakan salah satu alat manajemen sistem tenaga listrik yang digunakan untuk mengalokasikan kebutuhan pembangkitan tenaga listrik terhadap sejumlah unit pembangkit untuk memenuhi permintaan beban aktif. Dengan penerapan ED maka akan didapatkan biaya pembangkitan yang minimum terhadap produksi daya listrik yang dibangkitkan unit-unit pembangkit pada suatu sistem kelistrikan. Permasalahan ED pada pembangkit saat ini yaitu mengenai perubahan permintaan daya. Perubahan beban yang berubah setiap periode waktu tertentu menyebabkan perubahan perhitungan ED untuk setiap harga tertentu. Dari permasalahan tersebut, maka digunakan *dynamic formulation technique* untuk mengoptimalkan pengalokasian perubahan permintaan beban pada masing-masing unit pembangkitan serta meminimalkan biaya pembangkitan. Perhitungan ED menggunakan *dynamic formulation technique* dibandingkan dengan metode *base point and participation factor*. Tugas akhir ini mengembangkan program untuk melakukan perhitungan ED menggunakan metode *dynamic formulation technique* dan *base point and participation factor*. Dalam proses pengembangan program, digunakan software Delphi 7. Program yang telah dikembangkan di uji validitasnya sebelum digunakan untuk memecahkan studi kasus berdasarkan paper referensi. Dari hasil simulasi program, secara umum diperoleh hasil kombinasi biaya yang lebih murah dengan menggunakan *dynamic formulation technique* dibandingkan dengan *base point and participation factor*.

Kata kunci : Economic Dispatch, Dynamic Formulation Technique, Base Point and Participation Factor.

ABSTRACT

“Generator Load Optimization for Minimalizing Generation Cost Using Dynamic Formulation Technique”

Oktarina Ratri Wijayanti
2213106001

Advisor 1 : Prof. Ir. Ontoseno Penangsang, M.Sc, Ph.D
Advisor 2 : Dr. Rony Seto Wibowo, ST., MT.

Abstract:

Economic dispatch (ED) is one of the power system energy management tools that is used to allocate required power generation to a number of generating units to meet the active load demand. The ED's implementation can use to minimizing the generating cost of the generating units in power system. ED problems is about changes in the load demand. Change in the load at any specific time period causes changes in the calculation of ED for any specific price. Of these problems then the dynamic formulation technique used to optimally allocate the change in the total active load demand to the generating units. ED calculations using dynamic formulation technique instead of the base point and participation factor. This final project is about the developing of software to do ED calculation using dynamic formulation technique and base point and participation factor. The software developed using Delphi 7. After the program was built, firstly, it must be validated by validation test. The program need to be valid before processing study case that based on paper reference. ED's calculation using dynamic formulation technique achieve lower fuel generation cost compared with base point and participation method.

Kata kunci : Economic Dispatch, Dynamic Formulation Technique, Base Point and Participation Factor.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, karunia, dan petunjuk yang telah dilimpahkan-Nya sehingga penulis mampu menyelesaikan tugas akhir dengan judul :

“Optimasi Pembebanan Generator Untuk Meminimalkan Biaya Operasi Pembangkitan Menggunakan *Dynamic Formulation Technique*”

Tugas akhir ini disusun sebagai salah satu persyaratan untuk menyelesaikan jenjang pendidikan S1 pada Bidang Studi Teknik Sistem Tenaga, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Dalam penyusunan Tugas Akhir ini, penulis tidak lepas dari petunjuk, bimbingan, bantuan, dan dukungan dari berbagai pihak. Pada kesempatan ini penulis hendak menyampaikan rasa terima kasih kepada pihak-pihak yang telah membeir bantuan baik itu berupa moral maupun material, langsung maupun tidak langsung :

1. Allah SWT atas limpahan Rahmat dan Petunjuk-Nya serta Nabi Muhammad SAW atas tuntunan jalan-Nya.
2. Bapak dan Ibu yang telah membesarkan dan membimbing saya.
3. Prof. Ir. Ontoseno Penangsang, M.Sc, Ph.D dan Dr. Rony Seto Wibowo, ST., MT. sebagai dosen pembimbing yang telah memberikan arahan dan perhatiannya dalam Tugas Akhir ini.
4. Seluruh dosen yang telah memberikan ilmunya selama kuliah, karyawan, dan keluarga besar Jurusan teknik Elektro ITS
5. Teman-teman LJ Genap Teknik Elektro ITS 2013 dan khususnya kepada teman-teman satu kelompok TA atas bantuan kalian selama masa pengerjaan Tugas Akhir ini
6. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung, yang tidak mungkin saya sebutkan satu per satu

Untuk semuanya saya ucapkan terima kasih. Penulis menyadari bahwa Tugas Akhir ini belum sempurna, Oleh karena itu saran dan masukan sangat diharapkan untuk perbaikan dimasa yang akan datang.

Surabaya, Januari 2016

Penulis

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian.....	2
1.3 Permasalahan.....	2
1.4 Batasan Masalah.....	2
1.5 Metode Penelitian.....	2
1.6 Sistematika Penulisan.....	3
1.7 Manfaat dan Relevansi	4
BAB 2 ECONOMIC DISPATCH	5
2.1 Sistem Pembangkitan	5
2.2 Karakteristik Pembangkit Thermal.....	5
2.2.1 Pemodelan Fungsi Polinomial (Continuous)	7
2.2.2 Pemodelan Piecewise Incremental Heat.....	8
2.3 Economic Dispatch (ED).....	9
BAB 3 PENGAPLIKASIAN DYNAMIC FORMULATION TECHNIQUE PADA ECONOMIC DISPATCH.....	13
3.1 Algoritma Penyelesaian Tugas Akhir.....	13
3.2 Sintaksis Program <i>Dynamic Formulation Technique</i> Pada Delphi	16
3.3 Argumentasi Input Output Pada Delphi	17
3.4 Software Powergen	17
3.5 Profil Kasus Pengujian	21
3.5.1 Kasus 1	21
3.5.2 Kasus 2.....	23
3.5.3 Kasus 3.....	24

3.6 Perhitungan Manual Uji Validasi	28
3.6.1 Kasus 1a	28
3.6.1.1 Dynamic Formulation Technique	29
3.6.1.2 Base Point and Participation Factor	30
3.6.2 Kasus 1b	32
3.6.2.1 Dynamic Formulation Technique	32
3.6.2.2 Base Point and Participation Factor	33
3.6.3 Kasus 2a	34
3.6.3.1 Dynamic Formulation Technique	35
3.6.3.2 Base Point and Participation Factor	36
3.6.4 Kasus 2b	38
3.6.4.1 Dynamic Formulation Technique	38
3.6.4.2 Base Point and Participation Factor	38
BAB 4 HASIL SIMULASI DAN ANALISA.....	41
4.1 Validasi Program <i>Dynamic Formulation Technique</i>	41
4.1.1 Simulasi 1a	41
4.1.2 Simulasi 1b	43
4.1.3 Analisa	45
4.2 Sistem Dengan 6 Unit Pembangkit	45
4.2.1 Simulasi 2a	46
4.2.2 Simulasi 2b	48
4.2.3 Analisa	49
4.3 Sistem Dengan 18 Unit Pembangkit	50
4.3.1 Simulasi 3a	51
4.3.2 Simulasi 3b	54
4.3.3 Analisa	55
BAB 5 PENUTUP.....	59
5.1 Kesimpulan	59
5.2 Saran	59
DAFTAR PUSTAKA	61
LAMPIRAN.....	63
RIWAYAT HIDUP PENULIS.....	79

DAFTAR GAMBAR

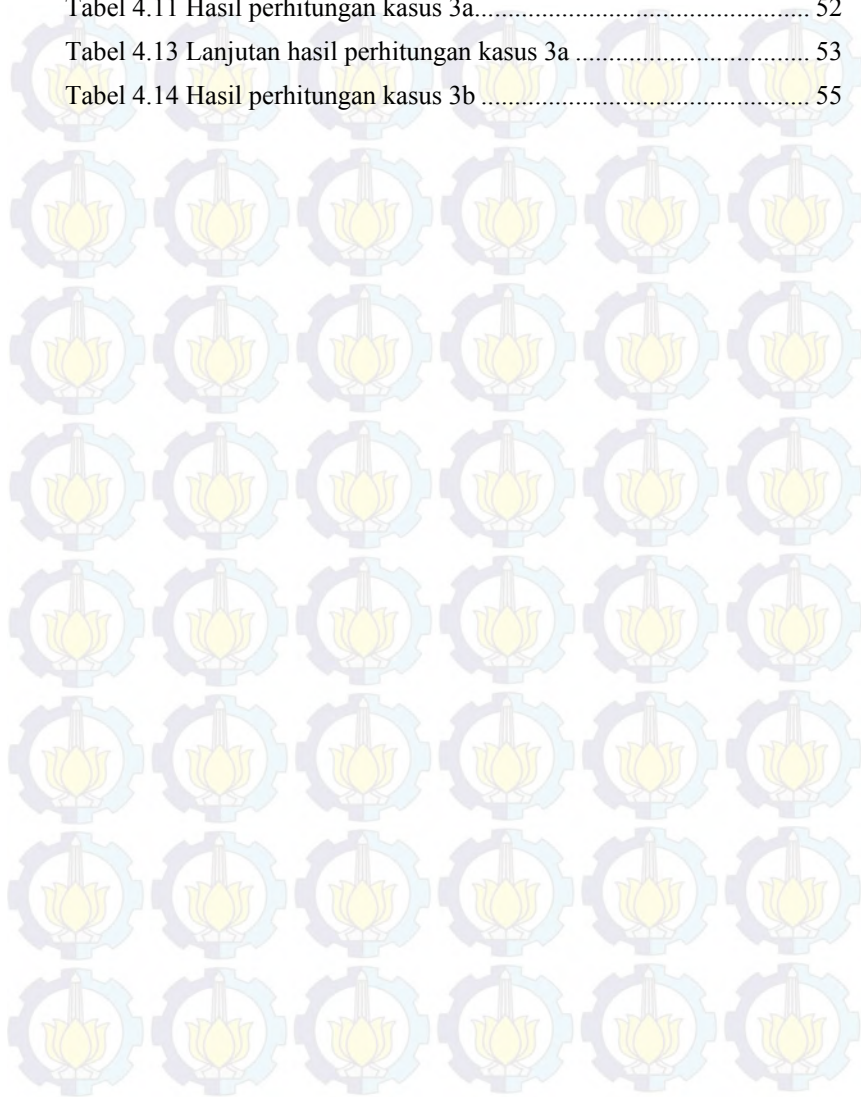
Gambar 2.1	Kurva input-output pembangkit thermal.....	6
Gambar 2.2	Kurva incremental pembangkit thermal	7
Gambar 2.3	Contoh kurva piecewise incremental rate.....	8
Gambar 3.1	Flowcart penerapan ED pada Delphi.....	13
Gambar 3.2	Flowcart penerapan <i>dynamic formulation technique</i> pada Delphi	14
Gambar 3.3	Tampilan Menu Utama <i>Software</i> Powergen.....	18
Gambar 3.4	Tampilan Utama Menu EDC.....	19
Gambar 3.5	Tampilan Pengisian Data Pembangkit.....	19
Gambar 3.6	Tampilan pilihan pengisian metode perhitungan <i>losses</i> .	20
Gambar 3.7	Tampilan <i>Set up Solution</i> menu EDC.....	20
Gambar 3.8	Tampilan hasil perhitungan EDC	21
Gambar 3.9	<i>Single line diagram</i> kasus 1.....	22
Gambar 3.10	<i>Single line diagram</i> kasus 2.....	23
Gambar 3.11	<i>Single line diagram</i> kasus 3.....	25
Gambar 4.1	Tampilan hasil perhitungan <i>dynamic formulation technique</i> kasus 1a	42
Gambar 4.2	Tampilan hasil perhitungan <i>base point and participation factor</i> kasus 1a.....	42
Gambar 4.3	Tampilan hasil perhitungan <i>dynamic formulation technique</i> kasus 1b	44
Gambar 4.4	Tampilan hasil perhitungan <i>base point and participation factor</i> kasus 1b.....	44
Gambar 4.5	Tampilan hasil perhitungan <i>dynamic formulation technique</i> kasus 2a	46
Gambar 4.6	Tampilan hasil perhitungan <i>base point and participation factor</i> kasus 2a.....	47

Gambar 4.7	Tampilan hasil perhitungan <i>base point and participation factor</i> kasus 2b.....	48
Gambar 4.7	Tampilan hasil perhitungan <i>dynamic formulation technique</i> kasus 3	51
Gambar 4.8	Tampilan hasil perhitungan <i>base point and participation factor</i> kasus 3a tanpa batas pembangkitan generator.....	51
Gambar 4.9	Tampilan hasil perhitungan <i>base point and participation factor</i> kasus 3a dengan batas pembangkitan generator...	51
Gambar 4.10	Tampilan hasil perhitungan <i>base point and participation factor</i> kasus 3b tanpa batas generator	54
Gambar 4.11	Tampilan hasil perhitungan <i>base point and participation factor</i> kasus 3b dengan batas generator	54

DAFTAR TABEL

Tabel 3.1	Daftar sintaksis yang digunakan pada program.....	16
Tabel 3.2	Lanjutan Daftar sintaksis yang digunakan pada program..	17
Tabel 3.3	Daftar argumentasi input output pada Delphi.....	17
Tabel 3.4	Data kasus 1	22
Tabel 3.5	Pembagian beban tiap bus pada kasus 1	23
Tabel 3.6	Data kasus 2	24
Tabel 3.7	Pembagian beban tiap bus pada kasus 2.....	24
Tabel 3.8	Data kasus 3	26
Tabel 3.9	Pembagian beban tiap bus pada kasus 3.....	26
Tabel 3.10	Lanjutan pembagian beban tiap bus pada kasus 3	27
Tabel 3.11	Lanjutan pembagian beban tiap bus pada kasus 3	28
Tabel 3.12	Hasil perhitungan ED menggunakan dynamic formulation technique	36
Tabel 3.13	Hasil perhitungan ED menggunakan <i>base point and participation factor</i> dengan beban awal 1150 MW	38
Tabel 3.14	Hasil perhitungan ED menggunakan <i>base point and participation factor</i> dengan beban awal 1250 MW	40
Tabel 4.1	Data kasus 1a.....	41
Tabel 4.2	Hasil perhitungan kasus 1a.....	43
Tabel 4.3	Lanjutan hasil perhitungan kasus 1a.....	43
Tabel 4.4	Hasil perhitungan kasus 1b.....	45
Tabel 4.5	Lanjutan hasil perhitungan kasus 1b	45
Tabel 4.6	Data kasus 2	46
Tabel 4.7	Hasil perhitungan kasus 2a.....	47
Tabel 4.8	Lanjutan hasil perhitungan kasus 2a.....	48
Tabel 4.9	Hasil perhitungan kasus 2b.....	49

Tabel 4.10 Data kasus 3	50
Tabel 4.11 Hasil perhitungan kasus 3a.....	52
Tabel 4.13 Lanjutan hasil perhitungan kasus 3a	53
Tabel 4.14 Hasil perhitungan kasus 3b	55



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Economic Dispatch (ED) merupakan salah satu alat manajemen sistem tenaga listrik yang digunakan untuk mengalokasikan kebutuhan pembangkitan tenaga listrik terhadap sejumlah unit pembangkit untuk memenuhi permintaan beban aktif. Ketika ada penambahan beban, tentu ada perubahan besarnya daya yang dibangkitkan dari masing-masing unit pembangkit namun tetap dengan biaya operasi yang minimum. Teknik optimasi digunakan untuk menyelesaikan masalah ED. Metode tersebut antara lain metode Newton, metode Gradient, metode *lambda iteration*, metode *base point and participation factor*, dimana fungsi biaya pembangkitan dinyatakan dalam *fungsi piece-wise linear*.

Metode yang digunakan dalam Tugas Akhir ini adalah *dynamic formulation technique*. *Dynamic formulation technique* dikembangkan untuk mengoptimalkan alokasi perubahan permintaan beban aktif ke unit pembangkitan.

Simulasi yang biasa digunakan adalah MATLAB untuk mendapatkan ED, akan tetapi tidak semua pengguna dapat menggunakan script yang berada pada MATLAB. Oleh karena itu pada tugas akhir ini diusulkan menggunakan Delphi. Delphi merupakan bahasa pemrograman berbasis Windows yang menyediakan fasilitas pembuatan aplikasi visual. Delphi memberikan kemudahan dalam menggunakan kode program, kompilasi yang cepat, penggunaan file unit ganda untuk pemrograman modular, pengembangan perangkat lunak, pola desain yang menarik serta diperkuat dengan bahasa pemrograman yang terstruktur dalam bahasa pemrograman Object Pascal. Delphi memiliki tampilan khusus yang didukung suatu lingkup kerja komponen Delphi untuk membangun suatu aplikasi dengan menggunakan Visual Component Library (VCL). Sebagian besar pengembang Delphi menuliskan dan mengkompilasi kode program dalam IDE (Integrated Development Environment). Pada tujuan tugas akhir ini akan mendapatkan aplikasi software perhitungan ED yang mudah untuk diinteraksikan dengan pengguna.

1.2 Tujuan Penelitian

Pembagian pembebanan generator dari hasil perhitungan menggunakan *dynamic formulation technique* lebih optimal daripada menggunakan metode *base point and participation factor*. Hal tersebut bisa dilihat dari besarnya biaya pembangkitan yang lebih minimum.

Mendapatkan aplikasi perhitungan ED dapat digunakan dengan mudah untuk berinteraksikan dengan pengguna. Serta mengalokasikan pendistribusian daya secara optimal pada masing-masing unit pembangkit saat ada penambahan beban dan meminimalkan biaya pembangkitan.

1.3 Permasalahan

Permasalahan yang akan dibahas dalam tugas akhir ini adalah :

1. Bagaimana menghitung ED suatu sistem pembebanan statis dengan menggunakan *dynamic formulation technique* dengan dibandingkan dengan *base point and participation factor*.
2. Bagaimana mengaplikasikan rumus matematis *dynamic formulation technique* ke dalam pemrograman Delphi.

1.4 Batasan Masalah

1. Sistem tidak memperhatikan rugi transmisi.
2. Tipe kurva input-output pembangkitan yang digunakan adalah tipe polinomial.
3. Metode *base point and participation factor* digunakan sebagai pembanding hasil dari *dynamic formulation technique*.
4. *Software* yang digunakan untuk dikembangkan adalah *software Delphi*.

1.5 Metode Penelitian

Alur metodologi penyelesaian tugas akhir ini adalah sebagai berikut:

1. Studi pustaka
Studi pustaka dilakukan untuk mengumpulkan buku-buku maupun jurnal yang berkaitan tentang topik tugas akhir yang dibahas. Pustaka-pustaka yang dikumpulkan mencakup *Economic Dispatch* serta buku pemrograman Delphi.

2. Pengenalan *software* dan eksperimen
Pengenalan *software* dilakukan dengan mempelajari *software* yang akan dikembangkan disertai melakukan eksperimen-eksperimen untuk mengetahui bagaimana cara kerja *software* tersebut.
3. Pengujian awal dan *troubleshooting* terhadap *bug* yang muncul
Pengujian awal dilakukan untuk mencari *bug*/kesalahan dalam *software*. Pengujian ini lebih mengutamakan kelancaran penggunaan *software* sehingga diuji dengan data-data sederhana namun beragam.
4. Pengujian akhir dan finalisasi *software*
Pengujian akhir dilakukan untuk memantapkan kembali kinerja *software* yang telah dikembangkan. Pada pengujian ini lebih mengutamakan kinerja *software* dalam melakukan perhitungan sehingga diuji dengan data-data yang lebih kompleks.
5. Pembuatan laporan tugas akhir
Melakukan penulisan laporan yang menunjukkan hasil akhir dari tugas akhir.

1.6 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini dibagi menjadi lima bab dengan masing-masing bab diuraikan sebagai berikut :

1. BAB 1 merupakan pendahuluan yang berisi latar belakang, permasalahan, tujuan, metodologi, batasan masalah dan sistematika penulisan.
2. BAB 2 berisi teori penunjang yang membahas tentang Sistem kelistrikan, *Economic Dispatch*
3. BAB 3 berisi tentang uraian perencanaan, pembuatan, dan implementasi kedalam *software* yang dikembangkan, selain itu juga dilakukan perhitungan manual untuk pengujian validasi program yang akan dirancang.
4. BAB 4 berisi tentang hasil pengujian perangkat lunak yang telah dirancang.
5. BAB 5 berisi tentang kesimpulan dan saran-saran dari pembuatan sampai pengimplementasian perangkat lunak.

1.7 Manfaat dan Relevansi

1. Bagi perusahaan listrik
Tugas akhir ini diharapkan dapat memberikan manfaat bagi perusahaan listrik dalam memutuskan pola pembangkitan yang dilakukan sehingga mendapatkan biaya pembangkitan yang lebih baik.
2. Bagi bidang ilmu pengetahuan dan mahasiswa lain
Tugas akhir ini diharapkan dapat membantu perkembangan ilmu pengetahuan dengan menjadi alat bantu perhitungan ED yang handal dan mudah digunakan.

BAB 2

ECONOMIC DISPATCH

2.1 Sistem Pembangkitan

Secara umum sistem pembangkitan merupakan kumpulan dari unit pembangkit tenaga listrik yang terdiri dari beberapa komponen utama seperti turbin dan generator. Pembangkit tenaga listrik digunakan untuk membangkitkan daya listrik yang kemudian didistribusikan kepada konsumen. Di dalam sebuah sistem pembangkit, beberapa generator dioperasikan secara paralel dan dihubungkan dengan bus dalam suatu sistem tenaga listrik guna menyediakan total daya yang diperlukan [3].

Pembangkit tenaga listrik dapat dibedakan menjadi beberapa jenis sesuai dengan bahan bakar yang digunakan. Salah satu diantaranya adalah pembangkit listrik tenaga panas atau *thermal*. Pembangkit tipe ini merupakan pembangkit listrik yang mayoritas digunakan untuk memenuhi beban harian atau *base load*.

Setiap pembangkit memiliki karakteristik unit pembangkit masing-masing. Karakteristik unit pembangkit meliputi karakteristik *input-output* pembangkit, dan karakteristik *incremental rate* [4]. Karakteristik tersebut diperoleh dari data-data seperti: desain generator; pabrik pembuat generator; data historis pengoperasian generator; maupun data percobaan. Karakteristik unit pembangkit digunakan dalam perhitungan biaya pembangkitan dari tiap unit pembangkit sehingga dapat dicapai nilai ekonomis atau nilai optimum.

Karakteristik *input-output* dari pembangkit *thermal* merupakan hubungan antara *input* berupa bahan bakar yang digunakan dengan *output* berupa daya yang dibangkitkan tiap pembangkit. *Input* bahan bakar dinyatakan dalam bentuk MBtu/h atau konsumsi energi sedangkan *output* daya dinyatakan dalam bentuk MW atau daya yang dibangkitkan.

Karakteristik *incremental rate* pembangkit *thermal* merupakan hubungan antara perubahan daya pembangkitan yang dihasilkan dengan konsumsi bahan bakar yang dibutuhkan. *Incremental rate* biasanya dinyatakan dengan satuan Btu/kWh.

2.2 Karakteristik Pembangkit Thermal

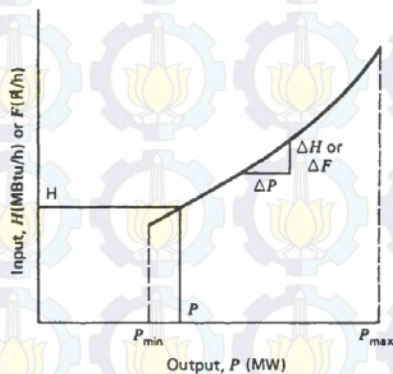
Ada banyak parameter dalam analisis pengaturan operasi sistem tenaga. Hal yang paling mendasar dalam masalah operasi ekonomis adalah karakteristik *input-output* pada pembangkit *thermal*. Untuk

menggambarkan karakteristik *input-output*, *input* merepresentasikan sebagai masukan total yang diukur dalam satuan biaya/jam dan *output* merupakan daya keluaran listrik yang disediakan oleh sistem pembangkit tenaga listrik. Dalam menggambarkan karakteristik unit turbin uap, akan menggunakan *termionologi* (2.1) dan (2.2) sebagai berikut:

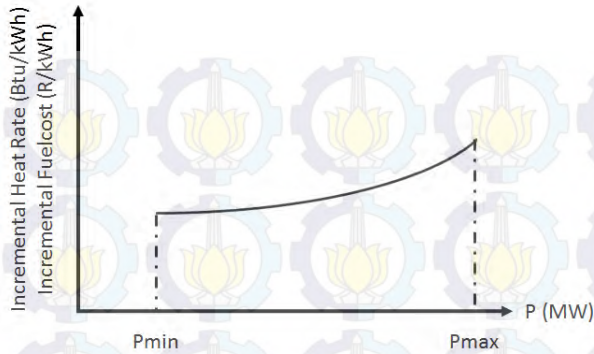
$$H = \frac{Mbtu}{jam} \quad (2.1)$$

$$F = \frac{R}{jam} \quad (2.2)$$

H dapat dinyatakan sebagai energi panas yang dibutuhkan tiap jam dan F dinyatakan sebagai biaya tiap jam. Ada kalanya R/jam biaya operasional suatu unit terdiri dari biaya operasional dan biaya pemeliharaan. Biaya karyawan akan dimasukkan sebagai bagian dari biaya operasi jika biaya ini dapat digambarkan secara langsung sebagai fungsi dari *output* unit. *Output* dari unit pembangkit dinyatakan dengan P dalam Megawatt. Untuk lebih jelasnya dapat dilihat pada Gambar 2.1 dan Gambar 2.2



Gambar 2.1 Kurva input-output pembangkit thermal[4]



Gambar 2.2 Kurva incremental pembangkit thermal[4]

Karakteristik *input-output* dari unit pembangkit *thermal* yang ideal, digambarkan sebagai kurva nonlinear yang kontinyu. Data karakteristik input output diperoleh dari perhitungan desain atau. Pembangkit *thermal* mempunyai batas operasi *minimum* (P_{min}) dan *maximum* (P_{max}). Batasan beban minimum biasanya disebabkan oleh kestabilan pembakaran dan masalah desain generator. Pada umumnya unit pembangkit thermal tidak dapat beroperasi dibawah 30% dari kapasitas desain.

Selanjutnya akan dibahas lebih lanjut tentang pemodelan karakteristik *input-output* maupun karakteristik *incremental rate*. Ada dua macam pendekatan dalam memodelkan karakteristik-karakteristik tersebut. Yang paling umum ditemui adalah pemodelan dengan fungsi polinomial. Namun disamping itu tidak jarang juga kita temui bentuk fungsi *piecewise*

2.2.1 Pemodelan Fungsi Polinomial (Continuous)

Bentuk pemodelan fungsi polinomial adalah pendekatan dari kurva *input-output* dengan fungsi polinomial. Fungsi polinomial yang umum digunakan adalah kurva polinomial orde dua. Namun meski begitu tidak menutup kemungkinan bila nantinya ada pendekatan dengan fungsi polinomial dengan orde lebih dari dua.

Sebagai contoh dari gambar 2.3 kita dapat membuat pendekatan fungsi polinomial orde dua seperti pada persamaan 2.3

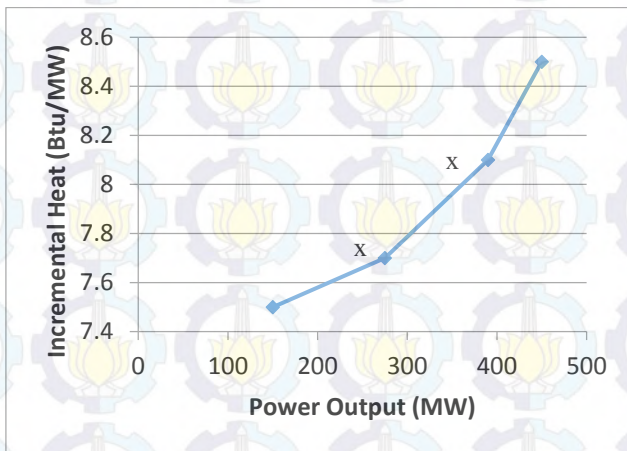
$$H(P) = aP^2 + bP + c \quad (2.3)$$

Sedangkan fungsi *incremental rate* nya bisa kita dapatkan dari turunan pertama fungsi *input-output*.

$$ihr(P) = \frac{\delta H(P)}{\delta P} = 2aP + b \quad (2.4)$$

2.2.2 Pemodelan Piecewise Incremental Heat

Dalam ilmu matematika, fungsi *piecewise* adalah fungsi yang didefinisikan oleh sub fungsi yang digunakan pada interval/segmen yang berbeda. Pemodelan bentuk ini menyajikan serangkaian set data dari kurva *incremental* heat yang kemudian dapat kita definisikan ke dalam bentuk polinomial untuk setiap interval/segmen. Penjelasannya akan lebih mudah jika kita mengamati Gambar 2.3



Gambar 2.3 Contoh kurva piecewise incremental rate

Untuk segmen antara titik (x_1, y_1) dan (x_2, y_2) dapat kita bentuk persamaan polinomialnya

$$ihr(P) = \alpha P + \beta \quad (2.5)$$

Dimana

$$\alpha = \frac{y_2 - y_1}{x_2 - x_1} \quad (2.6)$$

$$\beta = \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1} \quad (2.7)$$

Sehingga persamaannya menjadi

$$ihr(P) = \frac{y_2 - y_1}{x_2 - x_1} P + \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1} \quad (2.8)$$

Dari persamaan 2.6 kita bisa mendapatkan fungsi *input-output* nya dengan mengintegralkan fungsi *ihr*.

$$H = \int ihr(P) dp = \frac{1}{2} \alpha P^2 + \beta P + C \quad (2.9)$$

Dimana C adalah bahan bakar minimum saat output masih nol megawatt. C disebut juga *no load fuel*, atau pada fungsi biaya C disebut *no load cost*.

2.3 Economic Dispatch (ED)

Tingkat efisiensi dalam operasi optimalisasi ekonomi dan perencanaan daya pembangkitan listrik akan selalu menjadi bagian penting dalam perindustrian listrik. Oleh karena itu diperlukan perhitungan khusus akan pengiriman daya kepada para konsumen tenaga listrik sehingga perusahaan pemasok listrik tidak mengalami kerugian.

Tujuan utama dari *Economic Dispatch* (ED) adalah untuk menentukan kombinasi daya *output* yang minimal dari setiap unit pembangkit, dengan meminimalkan total biaya bahan bakar, sementara dapat memenuhi kebutuhan baban para konsumen. Pengoptimalan permasalahan ED sangat penting untuk melakukan perkiraan jangka panjang dalam sistem tenaga listrik, penentuan porsi biaya, dan pemodelan manajemen operasi tenaga listrik pada pembangkit.

Pembangkitan listrik memiliki tiga komponen biaya utama, antara lain biaya pembangunan, biaya kepemilikan, biaya operasional. Biaya operasional merupakan biaya yang berkaitan langsung dengan

keuntungan penjualan produksi. Hal ini dikarenakan biaya operasional berhubungan langsung dengan manajemen pembangkitan daya listrik.

Salah satu bagian yang paling penting dalam biaya operasional adalah biaya bahan bakar (*fuelcost*). Pada setiap unit pembangkitan nilai yang berbeda tergantung dari jenis bahan bakar yang digunakan dalam pembangkitan. Nilai dari *fuelcost* sangat mempengaruhi fungsi biaya yang didapat. Secara umum nilai dari *fuelcost* dapat dinyatakan dalam persamaan (2.10) berikut.

$$fuelcost = \frac{Fuelprice}{Rating Thermal} = \frac{R}{Mbtu} \quad (2.10)$$

Fuelcost merupakan harga persatuan panas dari bahan bakar, atau dapat dinyatakan sebagai konversi satuan panas ke satuan mata uang.

Pengaruh nilai *fuelcost* terhadap fungsi biaya dalam dilihat dalam persamaan objektif ED berikut,

$$Hi(Pi) = aiPi^2 + biPi + ci \quad (2.11)$$

$$i(Pi) = Hi(Pi) \times fuelcost \quad (2.12)$$

$$Ftotal = \min \sum_{i=1}^n Fi(Pi) \quad (2.13)$$

Dimana

n : jumlah generator

Dengan terhubungnya banyak unit pembangkit dalam sebuah sistem interkoneksi memberikan kemungkinan pengaturan pembangkitan yang lebih kecil untuk setiap unit.

Equality Constrain merupakan batasan yang merepresentasikan keseimbangan daya dalam sistem. Fungsi persamaan pada ED dinyatakan dalam persamaan,

$$\sum_{i=1}^n Pi = Pload + Ploss, n = \text{jumlah generator} \quad (2.14)$$

Inequality Constrain merupakan batasan yang merepresentasikan kapasitas daya dari pembangkit. Pada ED fungsi pertidaksamaan dinyatakan dalam persamaan (2.15) berikut.

$$Pi \min \leq Pi \leq Pi \max \quad (2.15)$$

Jika batasan *minimum* memiliki nilai seperti yang didapatkan pada persamaan (2.16) maka akan didapatkan solusi (2.17).

$$P_i \leq P_i \min \quad (2.16)$$

$$P_i = P_i \min \quad (2.17)$$

Jika batasan *maximum* memiliki nilai seperti yang didapatkan pada persamaan (2.18) maka akan didapatkan solusi (2.19).

$$P_i \geq P_i \max \quad (2.18)$$

$$P_i = P_i \max \quad (2.19)$$

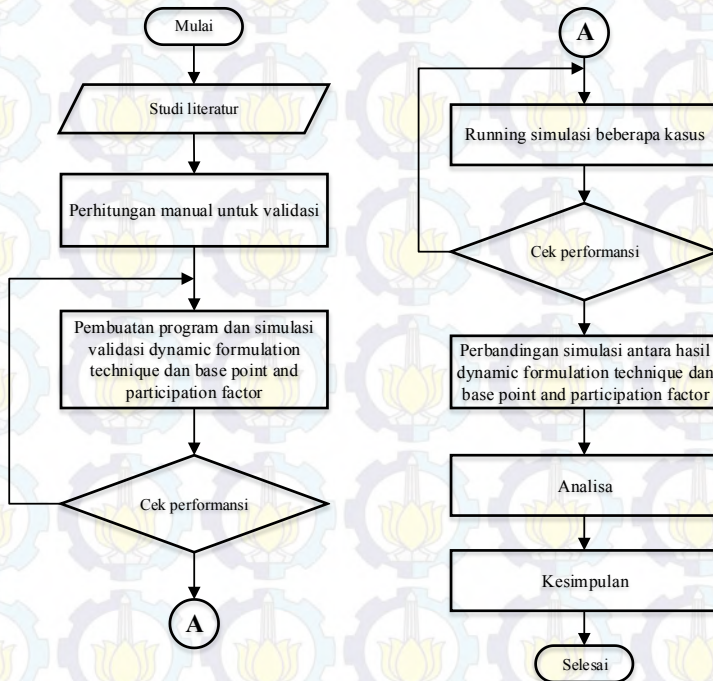


BAB 3

PENGAPLIKASIAN DYNAMIC FORMULATION TECHNIQUE PADA ECONOMIC DISPATCH

Dalam bab ini dijelaskan mengenai *dynamic formulation technique* untuk menyelesaikan permasalahan *economic dispatch* pada suatu sistem kelistrikan. Pengolahan data dan simulasi dikerjakan dengan menggunakan hasil pengembangan *software* Powergen yang berbasis Delphi 7.

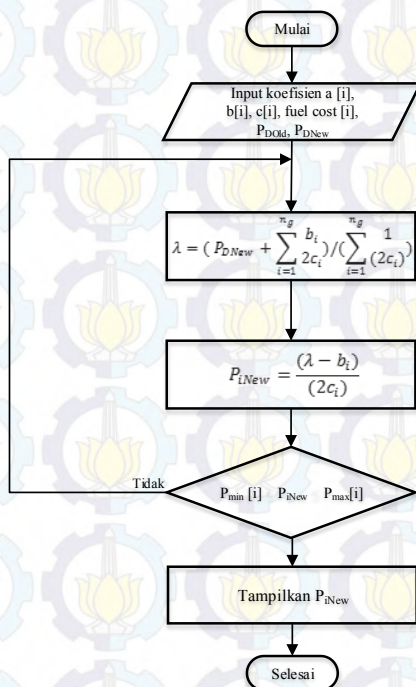
3.1 Algoritma Penyelesaian Tugas Akhir



Gambar 3.1 Flowcart penerapan ED pada Delphi

Alur dari penggunaan aplikasi perhitungan ED dimulai dengan mengumpulkan semua data yang dibutuhkan. Mulai dari jumlah unit, koefisien tiap orde, *fuelcost*, data batasan *maximum* dan *minimum* tiap unit. Setelah itu menentukan berapa beban yang ingin diperhitungkan. Perhitungan akan menggunakan *dynamic formulation technique* dan *base point and participation factor* untuk menentukan pembangkitan setiap unit. Besarnya pembangkitan masing-masing unit akan dimasukkan ke dalam persamaan fungsi biaya untuk mendapatkan nilai biaya pada beban.

Adapun flowchart penyelesaian ED menggunakan *dynamic formulation technique* seperti pada gambar 3.2 berikut.



Gambar 3.2 Flowcart penerapan *dynamic formulation technique* pada Delphi

Dynamic formulation technique merupakan suatu metode untuk mengalokasikan perubahan pada total kebutuhan beban aktif pada unit pembangkitan[5]. Asumsikan sebuah sistem tenaga listrik terdiri dari n

unit generator dengan total permintaan daya aktif beban P_D . Setiap unit generator i memiliki nilai pembangkitan optimal P_i . Perubahan pada daya yang dibangkitkan masing-masing generator i dipengaruhi oleh perubahan permintaan beban aktif ΔP_D yang di denotasikan dengan ΔP_i . Fungsi biaya pembangkitan untuk daya yang dibangkitkan $(P_i + \Delta P_i)$ adalah

$$F_i(P_i + \Delta P_i) = a_i + b_i(P_i + \Delta P_i) + c_i(P_i + \Delta P_i)^2 \quad (3.1)$$

$$\frac{\partial L}{\partial \Delta P_i} = \left(\frac{\partial F_i(P_i + \Delta P_i)}{\partial \Delta P_i} \right) - \lambda = 0 \quad (3.2)$$

Dimana λ adalah Lagrange multiplier atau Sistem peningkatan biaya. Perubahan pada total permintaan beban aktif ΔP_D adalah

$$\Delta P_D = \sum_{i=1}^{n_g} \Delta P_i \quad (3.3)$$

Dari persamaan (3.2) diperoleh

$$\left(\frac{\partial F_i(P_i + \Delta P_i)}{\partial \Delta P_i} \right) = \lambda \quad (3.4)$$

Dari persamaan (3.1) dan (3.3) diperoleh

$$\lambda = \left(\frac{\partial F_i(P_i + \Delta P_i)}{\partial \Delta P_i} \right) = (b_i + 2c_i(P_i + \Delta P_i)) \quad (3.5)$$

Jika dijumlahkan dari persamaan (3.5) untuk semua pembangkitan, diperoleh

$$\sum_{i=1}^{n_g} \frac{1}{(2c_i)} \frac{dF_i}{d\Delta P_i} - \sum_{i=1}^{n_g} \frac{(b_i + 2c_i P_i)}{2c_i} = \sum_{i=1}^{n_g} \Delta P_i \quad (3.6)$$

$$\lambda \sum_{i=1}^{n_g} \frac{1}{(2c_i)} = \Delta P_D + \sum_{i=1}^{n_g} \frac{(b_i + 2c_i P_i)}{2c_i} \quad (3.7)$$

Sehingga diperoleh nilai lambda dari persamaan (3.7) adalah

$$\lambda = (\Delta P_D + \sum_{i=1}^{n_g} P_i + \sum_{i=1}^{n_g} \frac{b_i}{2c_i}) / (\sum_{i=1}^{n_g} \frac{1}{(2c_i)}) \quad (3.8)$$

$$= (P_{DNew} + \sum_{i=1}^{n_g} \frac{b_i}{2c_i}) / (\sum_{i=1}^{n_g} \frac{1}{(2c_i)}) \quad (3.9)$$

Dimana $P_{DNew} = \Delta P_D + \sum_{i=1}^{n_g} P_i$ adalah total permintaan beban baru

$$\Delta P_i = \frac{(\lambda - (b_i + 2c_i P_i))}{(2c_i)} \quad (3.10)$$

Sehingga, daya pembangkitan optimal P_{iNew} untuk masing-masing unit i karena adanya total permintaan beban baru P_{DNew} adalah

$$P_{iNew} = \frac{(\lambda - b_i)}{(2c_i)} \quad (3.11)$$

3.2 Sintaksis Program *Dynamic Formulation Technique* Pada Delphi

Sintaksis program adalah perintah yang digunakan untuk melakukan pemanggilan program dengan argumen input yang kita masukkan. Sehingga untuk menjalankan metode gamma search pada program Powergen yang berbasis Delphi ini dibutuhkan sintaksis program sebagai berikut :

Tabel 3.1 Daftar sintaksis yang digunakan pada program

Sintaksis	Keterangan
datadump	Digunakan sebagai perintah menampilkan data permasalahan <i>ED</i> pada lembar <i>Output</i>
Data_input	Digunakan untuk menerima masukan dari data <i>file</i> yang telah tersimpan di awal
Data_output	Digunakan untuk menulis masukan pada data <i>file</i> , agar dapat tersimpan
ihr_ftn	Sebagai inisiasi turunan pertama persamaan $Hi(P_i(t))$ yang nantinya akan dikalikan dengan <i>fuelcost</i> , sehingga didapatkan inisiasi $\frac{\partial Fi}{\partial Pi}$
invers_ihr_ftn	Mencari nilai pembangkitan setiap unit ketika didapatkan nilai lambda

Tabel 3.2 Lanjutan Daftar sintaksis yang digunakan pada program

Sintaksis	Keterangan
prod_cost	Mendapatkan nilai biaya pembangkitan setiap unit setelah mendapatkan nilai pembangkitan yang optimal dari proses <i>lambdasearch</i>
base_point_dispatch	Sebagai prosedur penjalanan metode base point and participation factor untuk memperoleh nilai pembangkitan yang optimal
Dynamic_Technique_dispatch	Sebagai prosedur penjalanan metode dynamic formulation technique untuk memperoleh nilai pembangkitan yang optimal
Output_Routine	Merupakan prosedur untuk memperoleh hasil akhir tiap periode

3.3 Argumentasi Input Output Pada Delphi

Argumen input output adalah variabel yang dilibatkan sebagai data input dan output dalam program. Daftar argument input output bisa dilihat pada tabel 3.2.

Tabel 3.3 Daftar argumentasi input output pada Delphi

Argumen	Keterangan
Coeff[i,j]	Sebagai masukan awal dari nilai koefisien A, B, C dalam persamaan $Hi(Pi(t)) = ai + biPi(t) + ciPi(t)^2$
Fuelcost[i]	Sebagai masukan awal nilai dari <i>fuelcost</i> yang digunakan untuk persamaan $Fi(Pi(t)) = Hi(Pi(t)) \times fuelcosti$
Unitmax[i]	Sebagai masukan awal dari batas maximum pembangkitan unit (Pmax)
Unitmin[i]	Sebagaimasukan awal dari batasan minimum pembangititan unit (Pmin)

3.4 Software Powergen

Software Powergen adalah sebuah perangkat lunak milik Teknik Elektro Institut Teknologi Sepuluh Nopember yang digunakan untuk

melakukan perhitungan-perhitungan yang berkaitan dengan Teknik Sistem Tenaga. *Software* ini sudah digunakan sebagai alat bantu pada proses akademik. Salah satu mata kuliah yang menggunakan *software* ini adalah mata kuliah operasi optimum. Tampilan menu utama *software* Powergen dapat dilihat pada Gambar 3.3.



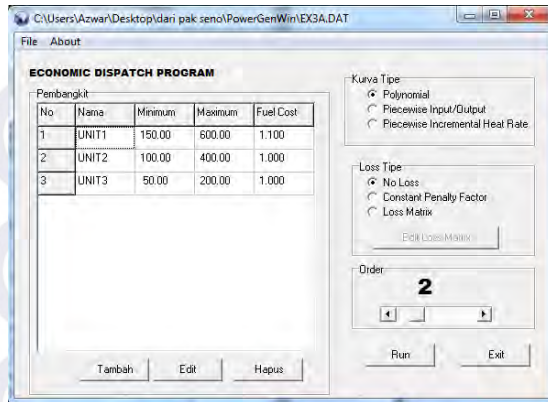
Gambar 3.3 Tampilan Menu Utama *Software* Powergen

Software Powergen yang akan dikembangkan memiliki beberapa fitur antara lain :

- Power Flow : Melakukan perhitungan aliran daya
- DUBLP : Melakukan perhitungan optimasi sederhana dengan *linear programming*
- EDC : Melakukan perhitungan *economic dispatch*
- Hydro : Menyelesaikan permasalahan penjadwalan pembangkit tenaga air/hydro
- Unitcom : Melakukan perhitungan *Unit Commitment*

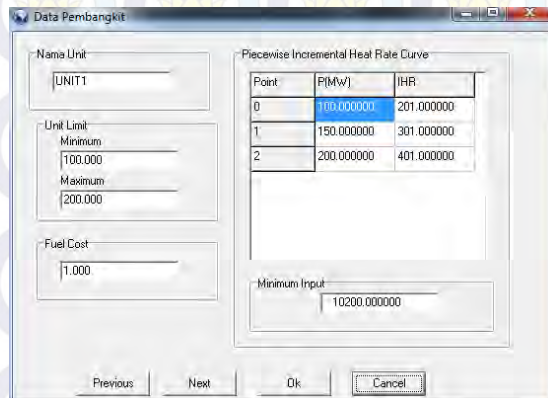
Pada tugas akhir ini menu yang akan dikembangkan adalah menu EDC (*Economic Dispatch*). Oleh karena itu akan dijelaskan terlebih dahulu fitur-fitur yang telah ada pada menu tersebut.

Menu EDC ini adalah menu pada program Powergen yang digunakan untuk melakukan perhitungan *economic dispatch*. Program Powergen ini mampu melakukan perhitungan *economic dispatch* hingga 20 unit pembangkit dengan tiga jenis tipe kurva. Tampilan menu EDC dapat dilihat pada Gambar 3.4.



Gambar 3.4 Tampilan Utama Menu EDC

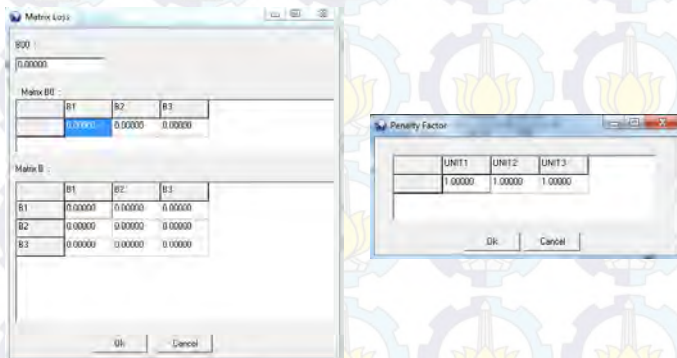
User/pengguna software mula mula mengisi data pembangkit dengan cara menekan tombol **Tambah** dan mengisi data-data pembangkit. *User* dapat mengisi kan data-data yang dimiliki oleh unit pembangkit seperti batasan pembangkitan minimum dan maksimum; karakteristik pembangkitan; serta biaya bahan bakar. Tampilan pengisian data-data pembangkit seperti pada Gambar 3.5.



Gambar 3.5 Tampilan Pengisian Data Pembangkit

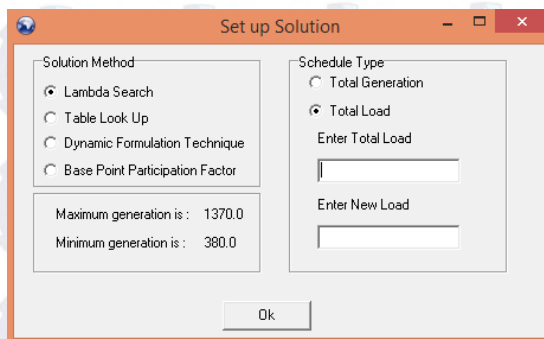
User pun dapat memilih metode perhitungan *losses* yang akan digunakan. Terdapat dua tipe perhitungan *losses* pada menu EDC ini jika

user ingin memasukkan faktor *losses* kedalam perhitungan *economic dispatch*-nya. Tipe perhitungan *losses* yang pertama adalah dengan matriks B_{loss} . Tipe perhitungan *losses* yang kedua adalah dengan *constant penalty factor*. Namun dalam tugas akhir ini *losses* yang ada akan diabaikan. Tampilan pengisian *losses* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Tampilan pilihan pengisian metode perhitungan *losses*

Setelah mengisi data-data yang dibutuhkan *user* dapat melanjutkan ke proses selanjutnya dengan memilih tombol **Run**. Ketika *user* memilih tombol **Run**, maka akan muncul tampilan seperti Gambar 3.7.

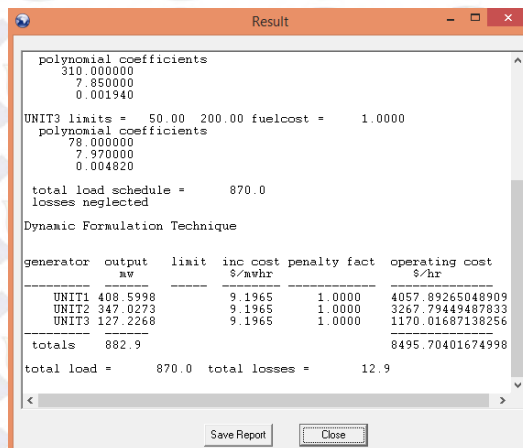


Gambar 3.7 Tampilan *Set up Solution* menu EDC

Pada tampilan ini *user* dapat memilihkan metode optimasi yang akan digunakan. Terdapat pilihan metode optimasi Lambda Search, Table

Look Up, Dynamic Formulation Technique, dan Base Point and Participation Factor. Pada tugas akhir ini yang akan dikembangkan hanya pada metode **Dynamic Formulation Technique dan Base Point and Participation Factor** dengan tipe kurva *polynomial*. Selain itu pada tampilan ini *user* dapat melihat pembangkitan maksimum dan pembangkitan minimum yang dapat dilakukan oleh unit-unit pembangkit yang ada. Selanjutnya *user* mengisikan **Total Generation** atau **Total Load** yang diinginkan sesuai **Schedule Type** yang dipilih.

Ketika semua data telah diisikan *user* dapat memilih **Ok** untuk menampilkan hasil perhitungan optimasi seperti Gambar 3.8.



Gambar 3.8 Tampilan hasil perhitungan EDC

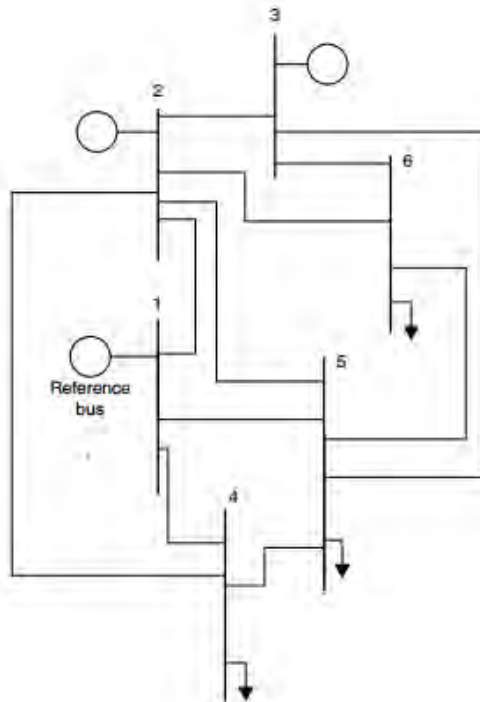
Pada tampilan hasil, *user* dapat memilih untuk menyimpan hasil perhitungan dengan menekan tombol **Save Report** atau dapat langsung menutup tampilan hasil dengan menekan tombol **Close**.

3.5 Profil Kasus Pengujian

3.5.1 Kasus 1

Kasus 1 merupakan kasus dengan 3 buah unit pembangkit dengan jumlah bus sebanyak 6 buah. Pada kasus 1, pembangkitan unit 1 pada bus 1 merupakan PLTU dengan bahan bakar batu bara. Pembangkitan unit 2 pada bus 2 dan 3 pada bus 3 merupakan PLTU dengan bahan bakar

minyak. Adapun *single line diagram* dari kasus 1 seperti pada gambar 3.9 dengan karakteristik pembangkit seperti pada tabel 3.4



Gambar 3.9 *Single line diagram* kasus 1

Tabel 3.4 Data kasus 1

Unit	P_i^{min}	P_i^{max}	a_i	b_i	c_i
1	100	600	561	7.92	0.001562
2	100	450	310	7.85	0.00194
3	50	200	78	7.97	0.00482

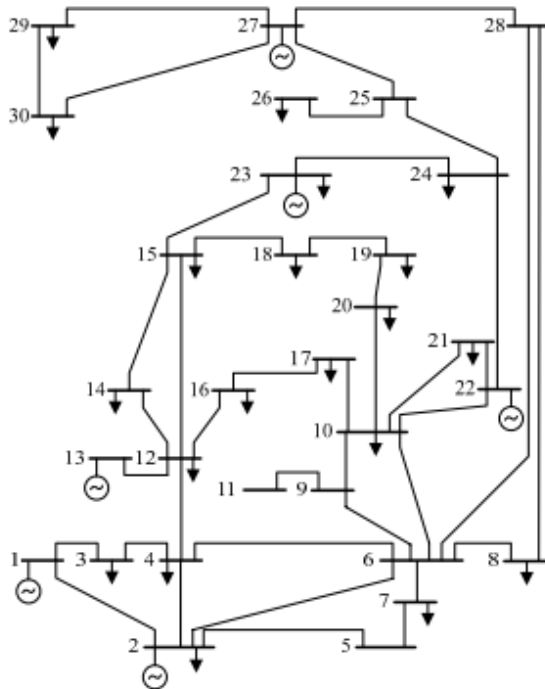
Pembagian beban pada tiap bus dengan total permintaan daya sebesar 850 MW per-jam bisa dilihat pada tabel 3.5

Tabel 3.5 Pembagian beban tiap bus pada kasus 1

Bus	Beban (MW)	Bus	Beban (MW)
1	0	4	250
2	0	5	250
3	100	6	250

3.5.2 Kasus 2

Kasus 2 merupakan kasus dengan 6 unit pembangkit dengan jumlah bus sebanyak 30 buah. Unit pembangkitan 1, 2, dan 3 merupakan PLTU dengan bahan bakar batu bara. Sedangkan unit pembangkitan 4, 5, dan 6 merupakan PLTU dengan bahan bakar minyak. Adapun *single line diagram* dari kasus 1 seperti pada gambar 3.10 dengan karakteristik pembangkit seperti pada tabel 3.6



Gambar 3.10 Single line diagram kasus 2

Tabel 3.6 Data kasus 2

Generator	Bus	P_i^{max}	P_i^{min}	a_i	b_i	c_i
1	1	500	100	240	7	0.007
2	2	200	50	200	10	0.0095
3	13	300	80	220	8	0.009
4	22	150	50	200	11	0.009
5	23	200	50	220	10.5	0.008
6	27	120	50	190	12	0.0075

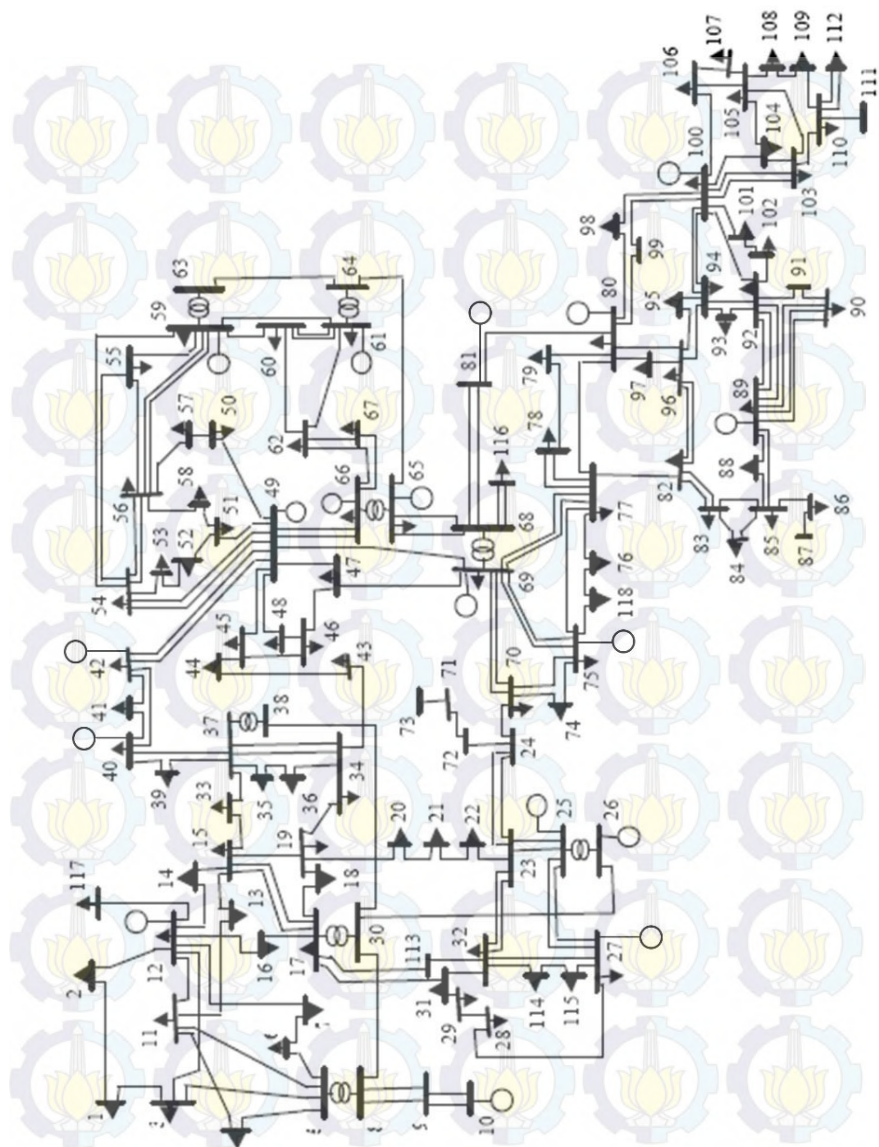
Pembagian beban pada tiap bus untuk kasus 2 dengan total permintaan beban aktif sebesar 1300 MW per-jam bisa dilihat pada tabel 3.7.

Tabel 3.7 Pembagian beban tiap bus pada kasus 2

Bus	Beban (MW)	Bus	Beban (MW)
1	0	16	15.5
2	82.8	17	45
3	10	18	11
4	313	19	42
5	162	20	9
6	0	21	76
7	112	22	0
8	141	23	15
9	0	24	43.5
10	26	25	0
11	0	26	17.5
12	50	27	0
13	0	28	0
14	30	29	10.7
15	40	30	48

3.5.3 Kasus 3

Kasus 3 merupakan kasus dengan jumlah pembangkit sebanyak 18 unit pembangkitan dengan jumlah bus sebanyak 118 buah. Unit pembangkitan 1, 2, 3, 4, 10, 11, 12, 13, 14, 15, dan 16 merupakan PLTU dengan bahan bakar batu bara. Sedangkan unit pembangkitan 5, 6, 7, 8, 9, 17, dan 18 merupakan PLTU dengan bahan bakar minyak. Adapun *single line diagram* dari kasus 3 seperti pada gambar 3.11 dengan karakteristik pembangkit seperti pada tabel 3.8.



Gambar 3.11 Single line diagram kasus 3

Tabel 3.8 Data kasus 3

Unit	Bus	p_i^{min}	p_i^{max}	a_i	b_i	c_i
1	10	7	15.00	85.74158	22.45526	0.602842
2	12	7	45.00	85.74158	22.45526	0.602842
3	25	13	25.00	108.9837	22.52789	0.214263
4	26	16	25.00	49.06263	26.75263	0.077837
5	27	16	25.00	49.06263	26.75263	0.077837
6	40	3	14.75	677.73000	80.39345	0.734763
7	42	3	14.75	677.73000	80.39345	0.734763
8	49	3	12.28	44.39000	13.19474	0.514474
9	59	3	12.28	44.39000	13.19474	0.514474
10	61	3	12.28	44.39000	13.19474	0.514474
11	65	3	12.28	44.39000	13.19474	0.514474
12	69	3	24.00	574.96030	56.70947	0.657079
13	75	3	16.20	820.37760	84.67579	1.236474
14	80	3	36.20	603.02370	59.59026	0.394571
15	81	3	45.00	567.93630	56.70947	0.420789
16	89	3	37.00	567.93630	55.96500	0.420789
17	100	3	45.00	567.93630	55.96500	0.420789
18	118	3	16.20	820.37760	84.67579	1.236474

Pembagian beban pada tiap bus dengan total permintaan daya sebesar 303.3 MW per-jam bisa dilihat pada tabel 3.9

Tabel 3.9 Pembagian beban tiap bus pada kasus 3

Bus	Beban (MW)	Bus	Beban (MW)
1	5	60	6
2	1.6	61	0
3	3.7	62	2.7
4	0	63	5.9
5	0	64	0
6	4.6	65	0
7	2	66	2.7
8	0	67	1.7
9	0	68	0
10	0	69	0
11	6.6	70	6

Tabel 3.10 Lanjutan pembagian beban tiap bus pada kasus 3

Bus	Beban (MW)	Bus	Beban (MW)
12	4	71	0
13	3	72	0
14	11.1	73	0
15	7.7	74	5.5
16	1.8	75	3.4
17	0.5	76	5.6
18	5.7	77	5.5
19	4	78	6
20	1	79	3.1
21	0.8	80	0
22	0.5	81	9.4
23	5.7	82	3.7
24	4.6	83	1.6
25	0	84	0.8
26	0	85	1.8
27	0	86	1.6
28	0.9	87	0
29	1.8	88	3.3
30	0	89	2.4
31	3.8	90	5.5
32	5.7	91	0
33	1.9	92	5.6
34	5.7	93	0.4
35	2.7	94	2.2
36	2.5	95	3.5
37	0	96	3.1
38	7.7	97	1
39	1.7	98	2.7
40	1.6	99	0
41	3	100	3.1
42	3	101	1.8
43	1	102	0
44	1	103	2
45	4.8	104	3.1
46	2	105	2.7

Tabel 3.11 Lanjutan pembagian beban tiap bus pada kasus 3

Bus	Beban (MW)	Bus	Beban (MW)
47	3	106	3.6
48	1.5	107	2.2
49	0	108	0
50	1.3	109	0
51	1.2	110	3.2
52	1	111	0
53	1.8	112	1.7
54	8.6	113	0
55	5.8	114	0.7
56	7.7	115	2
57	0.6	116	0
58	0.6	117	1.7
59	12	118	3

3.6 Perhitungan Manual Uji Validasi

Kasus pengujian digunakan untuk menguji kebenaran perhitungan yang dilakukan oleh *software*. Dalam tugas akhir ini akan ada beberapa kasus pengujian. Kasus 1 merupakan kasus yang digunakan untuk pengujian awal *software*. Pengujian awal bertujuan untuk menguji kelancaran dan ketepatan program. Sedangkan kasus 2 dan 3 merupakan kasus yang digunakan untuk pengujian akhir *software*. Pengujian akhir bertujuan untuk memperoleh perbandingan perhitungan *dynamic formulation technique* dan *base point and participation factor*.

Pada kasus 1 data pengujian diambil dari jurnal [5]. Kasus 1 akan dibagi menjadi 2 bagian yaitu 1a dengan permintaan beban baru 870 MW dan 1b dengan permintaan beban baru 900 MW. Sedangkan kasus 2 diambil dari jurnal [6].

3.6.1 Kasus 1a

Berikut adalah data pengujian awal kasus pertama. Kasus ini menggunakan tipe kurva polynomial orde dua dengan permintaan beban awal 850 MW dan permintaan beban baru 870 MW.

3.6.1.1 Dynamic Formulation Technique

$$\begin{aligned}\lambda &= (P_{DNew} + \sum_{i=1}^{n_g} \frac{b_i}{2c_i}) / (\sum_{i=1}^{n_g} \frac{1}{(2c_i)}) \\&= \frac{870 + \frac{7.92}{2 \times 0.001562} + \frac{7.85}{2 \times 0.00194} + \frac{7.97}{2 \times 0.00482}}{\frac{1}{(2 \times 0.001562)} + \frac{1}{2 \times 0.00194} + \frac{1}{2 \times 0.00482}} \\&= \frac{870 + 2535.2113 + 2023.1959 + 826.7635}{320.1024 + 257.732 + 103.7344} \\&= \frac{6255.1706}{681.5688} \\&\lambda = 9.177607 \text{ \$/MWh}\end{aligned}$$

Setelah didapat nilai λ , maka daya yang dibangkitkan masing-masing generator dapat dihitung seperti berikut :

$$\begin{aligned}P_{1New} &= \frac{(9.177607 - 7.92)}{2 \times 0.001562} = 402.5629 \text{ MW} \\P_{2New} &= \frac{(9.177607 - 7.85)}{2 \times 0.00194} = 342.1667 \text{ MW} \\P_{3New} &= \frac{(9.177607 - 7.97)}{2 \times 0.00482} = 125.2704 \text{ MW}\end{aligned}$$

Sehingga fungsi biaya pembangkitan untuk masing-masing unit pembangkit adalah :

$$\begin{aligned}F_1 &= 561 + (7.92)(402.5629) + (0.001562)(402.5629)^2 \\&= 4002.431 \text{ \$/jam} \\F_2 &= 310 + (7.85)(342.1667) + (0.00194)(342.1667)^2 \\&= 3223.140 \text{ \$/jam} \\F_3 &= 78 + (7.97)(125.2704) + (0.00482)(125.2704)^2 \\&= 1152.0438 \text{ \$/jam}\end{aligned}$$

$$F_1 + F_2 + F_3 = 8377.6148 \text{ \$/jam}$$

Total biaya pembangkitan adalah 8377.615 \\$/jam

3.6.1.2 Base Point and Participation Factor

Pada metode base point and participation factor, langkah pertama adalah menentukan P_{base} sebagai base point dari metode ini. P_{base} diperoleh dari hasil perhitungan menggunakan iterasi lambda.

Menentukan nilai lambda

$$\lambda_{min} = \min \left(n \frac{\partial F_i}{\partial P_i} P_f, i = 1 \dots \text{Jumlah Generator} \right)$$

$$\lambda_{min} = 8.2324$$

$$\lambda_{max} = \max \left(n \frac{\partial F_i}{\partial P_i} P_f, i = 1 \dots \text{Jumlah Generator} \right)$$

$$\lambda_{max} = 9.8980$$

Menentukan lambda start:

$$\lambda_{start} = \frac{\lambda_{max} + \lambda_{min}}{2} = 9.0652$$

Menentukan Delta Lambda (dicari untuk melakukan iterasi):

$$\Delta\lambda = \frac{\lambda_{max} - \lambda_{min}}{2} = 0.8328$$

Dengan memasukkan nilai lambda yang baru, kemudian melakukan iterasi pada lambda dengan cara menambah dan mengurangi nilai lambda sebesar $\frac{\Delta\lambda}{2}$. Hal ini bertujuan untuk mendapatkan hasil pembangkitan daya yang sesuai dengan load yang diinginkan. Dalam kasus ini permintaan beban awal yang diinginkan sebesar 850 MW.

$$\lambda_{iter1} = 9.0652 \dots \dots \dots totalgen = 793.4$$

$$\lambda_{iter2} = 9.8980 \dots \dots \dots totalgen = 1200$$

$$\lambda_{iter3} = 9.4816 \dots \dots \dots totalgen = 1056.7$$

$$\lambda_{iter4} = 9.2734 \dots \dots \dots totalgen = 935.3$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\lambda_{iter15} = 9.1483 \dots \dots \dots totalgen = 850.0$$

$$P_1 = \frac{\lambda - 7.92}{2 \times 0.001562} = \frac{9.1483 - 7.92}{2 \times 0.001562}$$

$$100 \leq P_1 \leq 600$$

$$P_1 = 393.168 \text{ MW}$$

$$P_2 = \frac{\lambda - 7.85}{2 \times 0.00194} = \frac{9.1483 - 7.85}{2 \times 0.00194}$$

$$100 \leq P_2 \leq 400$$

$$P_2 = 334.602 \text{ MW}$$

$$P_3 = \frac{\lambda - 7.97}{2 \times 0.00482} = \frac{9.1483 - 7.97}{2 \times 0.00482}$$

$$50 \leq P_3 \leq 200$$

$$P_3 = 122.226 \text{ MW}$$

Participation Factor Untuk Masing-masing unit

$$\left(\frac{\Delta P_i}{\Delta P_D} \right) = \frac{\left(\frac{1}{F_i} \right)}{\sum_i \left(\frac{1}{F_i} \right)}$$

$$\left(\frac{\Delta P_1}{\Delta P_D} \right) = \frac{\frac{1}{2 \times 0.001562}}{\left(\frac{1}{2 \times 0.001562} + \frac{1}{2 \times 0.00194} + \frac{1}{2 \times 0.00482} \right)} = 0.47$$

$$\left(\frac{\Delta P_2}{\Delta P_D} \right) = \frac{\frac{1}{2 \times 0.00194}}{\left(\frac{1}{2 \times 0.001562} + \frac{1}{2 \times 0.00194} + \frac{1}{2 \times 0.00482} \right)} = 0.38$$

$$\left(\frac{\Delta P_3}{\Delta P_D} \right) = \frac{\frac{1}{2 \times 0.00482}}{\left(\frac{1}{2 \times 0.001562} + \frac{1}{2 \times 0.00194} + \frac{1}{2 \times 0.00482} \right)} = 0.15$$

$$\Delta P_D = 870 - 850 = 20 \text{ MW}$$

Sehingga didapat daya pembangkitan yang baru akibat perubahan permintaan beban adalah

$$P_{iNew} = P_{iBase} + \left(\frac{\Delta P_i}{\Delta P_D} \right) \Delta P_D$$

$$P_{1New} = 393.168 + (0.4696)(20) = 402.5631 \text{ MW}$$

$$P_{2New} = 334.602 + (0.3781)(20) = 342.1629 \text{ MW}$$

$$P_{3New} = 122.226 + (0.1521)(20) = 125.2740 \text{ MW}$$

Setelah diperoleh besarnya daya pembangkitan masing-masing unit generator, fungsi biaya dari masing-masing unit pembangkitan dapat dihitung

$$F_1 = 561 + (7.92)(402.5631) + (0.001562)(402.5631)^2 \\ = 4002.4329 \text{ \$/jam}$$

$$F_2 = 310 + (7.85)(342.1629) + (0.00194)(342.1629)^2 \\ = 3223.1052 \text{ \$/jam}$$

$$F_3 = 78 + (7.97)(125.274) + (0.00482)(125.274)^2 \\ = 1152.0767 \text{ \$/jam}$$

$$F_1 + F_2 + F_3 = 8377.6148 \text{ \$/jam}$$

Sehingga total biaya pembangkitan adalah 8377.6148 \\$/jam

3.6.2 Kasus 1b

Kasus ini sama dengan kasus 1a, hanya saja jika pada kasus 1a permintaan beban baru adalah 870 MW, maka pada 1b permintaan beban baru adalah 900 MW

3.6.2.1 Dynamic Formulation Technique

$$\lambda = \left(P_{DNew} + \sum_{i=1}^{n_g} \frac{b_i}{2c_i} \right) / \left(\sum_{i=1}^{n_g} \frac{1}{(2c_i)} \right)$$

$$\lambda = \frac{900 + \frac{7.92}{2 \times 0.001562} + \frac{7.85}{2 \times 0.00194} + \frac{7.97}{2 \times 0.00482}}{\left(\frac{1}{2 \times 0.001562} + \frac{1}{2 \times 0.00194} + \frac{1}{2 \times 0.00482} \right)}$$

$$\begin{aligned}
&= \frac{900 + 2535.2113 + 2023.1959 + 826.7635}{320.1024 + 257.732 + 103.7344} \\
&= \frac{6285.1706}{681.5688} \\
\lambda &= 9.22162 \text{ \$/MWh}
\end{aligned}$$

Setelah didapat nilai λ , maka daya yang dibangkitkan masing-masing generator dapat dihitung seperti berikut :

$$P_{1New} = \frac{(9.22162 - 7.92)}{2 \times 0.001562} = 416.6526 \text{ MW}$$

$$P_{2New} = \frac{(9.22162 - 7.85)}{2 \times 0.00194} = 353.5110 \text{ MW}$$

$$P_{3New} = \frac{(9.22162 - 7.97)}{2 \times 0.00482} = 129.8364 \text{ MW}$$

Sehingga fungsi biaya pembangkitan untuk masing-masing unit pembangkit adalah :

$$\begin{aligned}
F_1 &= 561 + (7.92)(416.6526) + (0.001562)(416.6526)^2 \\
&= 4132.0508 \text{ \$/jam}
\end{aligned}$$

$$\begin{aligned}
F_2 &= 310 + (7.85)(353.511) + (0.00194)(353.511)^2 \\
&= 3327.5032 \text{ \$/jam}
\end{aligned}$$

$$\begin{aligned}
F_3 &= 78 + (7.97)(129.8364) + (0.00482)(129.8364)^2 \\
&= 1194.0492 \text{ \$/jam}
\end{aligned}$$

$$F_1 + F_2 + F_3 = 8653.6032 \text{ \$/jam}$$

Total biaya pembangkitan untuk seluruh unit pembangkitan adalah 8653.6032 \\$/jam

3.6.2.2 Base Point and Participation Factor

Karena pada kasus 1b nilai permintaan beban awal sama dengan kasus 1a, maka hasil dari perhitungan base point dan participation factor juga memiliki nilai yang sama dengan kasus 1a yaitu :

$$P_1 = 393.168 \text{ MW}$$

$$P_2 = 334.602 \text{ MW}$$

$$P_3 = 122.226 \text{ MW}$$

$$\Delta P_D = 900 - 850 = 50 \text{ MW}$$

Sehingga daya pembangkitan untuk masing-masing unit pembangkit adalah

$$P_{iNew} = P_{iBase} + \left(\frac{\Delta P_i}{\Delta P_D} \right) \Delta P_D$$

$$P_{1New} = 393.168 + (0.4696)(50) = 416.6528 \text{ MW}$$

$$P_{2New} = 334.602 + (0.3781)(50) = 353.5073 \text{ MW}$$

$$P_{3New} = 122.226 + (0.1521)(50) = 129.8400 \text{ MW}$$

Sehingga fungsi biaya pembangkitan untuk masing-masing unit pembangkit adalah :

$$\begin{aligned} F_1 &= 561 + (7.92)(416.6528) + (0.001562)(416.6528)^2 \\ &= 4132.0524 \text{ \$/jam} \end{aligned}$$

$$\begin{aligned} F_2 &= 310 + (7.85)(353.5073) + (0.00194)(353.5073)^2 \\ &= 3327.4690 \text{ \$/jam} \end{aligned}$$

$$\begin{aligned} F_3 &= 78 + (7.97)(129.84) + (0.00482)(129.84)^2 \\ &= 1194.0822 \text{ \$/jam} \end{aligned}$$

$$F_1 + F_2 + F_3 = 8653.6036 \text{ \$/jam}$$

Total biaya pembangkitan untuk seluruh unit pembangkit adalah 8653.6036 \\$/jam

3.6.3 Kasus 2a

Berikut adalah data pengujian awal kasus kedua. Kasus ini menggunakan tipe kurva polynomial orde dua dengan permintaan beban awal 1150 MW dan permintaan beban baru 1300 MW.

3.6.3.1 Dynamic Formulation Technique

$$\lambda = (P_{DNew} + \sum_{i=1}^{n_g} \frac{b_i}{2c_i}) / (\sum_{i=1}^{n_g} \frac{1}{(2c_i)})$$

$$\lambda = \frac{1300 + \frac{7}{2 \times 0.007} + \frac{10}{2 \times 0.0095} + \frac{8}{2 \times 0.009} + \frac{11}{2 \times 0.009} + \frac{10.5}{2 \times 0.008} + \frac{12}{2 \times 0.0075}}{\frac{1}{2 \times 0.007} + \frac{1}{2 \times 0.0095} + \frac{1}{2 \times 0.009} + \frac{1}{2 \times 0.009} + \frac{1}{2 \times 0.008} + \frac{1}{2 \times 0.0075}}$$

$$\lambda = \frac{1300 + 3565.12134}{364.3379} = 13.2792 \text{ \$/MWh}$$

Setelah didapat nilai λ , maka daya yang dibangkitkan masing-masing generator dapat dihitung seperti berikut :

$$P_{1New} = \frac{(13.2792 - 7)}{2 \times 0.007} = 448.515 \text{ MW}$$

$$P_{2New} = \frac{(13.2792 - 10)}{2 \times 0.0095} = 172.59 \text{ MW}$$

$$P_{3New} = \frac{(13.2792 - 8.5)}{2 \times 0.009} = 293.29 \text{ MW}$$

$$P_{4New} = \frac{(13.2792 - 11)}{2 \times 0.009} = 126.623 \text{ MW}$$

$$P_{5New} = \frac{(13.2792 - 10.5)}{2 \times 0.008} = 173.701 \text{ MW}$$

$$P_{6New} = \frac{(13.2792 - 12)}{2 \times 0.0075} = 85.2809 \text{ MW}$$

Setelah diperoleh besarnya daya pembangkitan masing-masing unit generator, fungsi biaya dari masing-masing unit pembangkitan adalah

Tabel 3.12 Hasil perhitungan ED menggunakan dynamic formulation technique

Generator	$P_{iNew} (MW)$	$F_i (\$/jam)$
1	448.515	4787.769
2	172.59	2208.882
3	293.29	3340.487
4	126.623	1737.154
5	173.701	2285.235
6	85.2809	1267.918
Total	1300	15627.44

3.6.3.2 Base Point and Participation Factor

Menentukan nilai lambda

$$\lambda_{min} = 8.4$$

$$\lambda_{max} = 14$$

$$\lambda_{start} = \frac{\lambda_{max} + \lambda_{min}}{2} = 11.2$$

$$\Delta\lambda = \frac{\lambda_{max} - \lambda_{min}}{2} = 2.8$$

$$\lambda_{iter1} = 11.2 \dots \dots \dots totalgen = 690.9$$

$$\lambda_{iter2} = 14 \dots \dots \dots totalgen = 1470$$

$$\lambda_{iter3} = 12.6 \dots \dots \dots totalgen = 1062.5$$

$$\lambda_{iter4} = 13.3 \dots \dots \dots totalgen = 1307.6$$

⋮

$$\lambda_{iter18} = 12.8675 \dots \dots \dots totalgen = 1150$$

$$P_1 = \frac{12.8675 - 7}{2 \times 0.007} = 419.1071 \text{ MW}$$

$$P_2 = \frac{12.8675 - 10}{2 \times 0.0095} = 150.9211 \text{ MW}$$

$$P_3 = \frac{12.8675 - 8}{2 \times 0.009} = 270.4167 \text{ MW}$$

$$P_4 = \frac{12.8675 - 11}{2 \times 0.009} = 103.75 \text{ MW}$$

$$P_5 = \frac{12.8675 - 10.5}{2 \times 0.008} = 147.9688 \text{ MW}$$

$$P_6 = \frac{12.8675 - 12}{2 \times 0.0075} = 57.8333 \text{ MW}$$

Participation Factor Untuk Masing-masing unit

$$\left(\frac{\Delta P_i}{\Delta P_D} \right) = \frac{\left(\frac{1}{F_i} \right)}{\sum_i \left(\frac{1}{F_i} \right)}$$

$$\left(\frac{\Delta P_1}{\Delta P_D} \right) = \frac{142.8571}{728.6759} = 0.196$$

$$\left(\frac{\Delta P_2}{\Delta P_D} \right) = \frac{105.2632}{728.6759} = 0.144$$

$$\left(\frac{\Delta P_3}{\Delta P_D} \right) = \frac{111.1111}{728.6759} = 0.152$$

$$\left(\frac{\Delta P_4}{\Delta P_D} \right) = \frac{111.1111}{728.6759} = 0.152$$

$$\left(\frac{\Delta P_5}{\Delta P_D} \right) = \frac{125}{728.6759} = 0.172$$

$$\left(\frac{\Delta P_6}{\Delta P_D} \right) = \frac{133.333}{728.6759} = 0.183$$

$$\Delta P_D = 1300 - 1150 = 150$$

$$P_{iNew} = P_{iBase} + \left(\frac{\Delta P_i}{\Delta P_D} \right) \Delta P_D$$

$$P_{1New} = 419.1071 + (0.196)(150) = 448.515 \text{ MW}$$

$$P_{2New} = 150.9211 + (0.144)(150) = 172.59 \text{ MW}$$

$$P_{3New} = 270.4167 + (0.152)(150) = 293.289 \text{ MW}$$

$$P_{4New} = 103.75 + (0.152)(150) = 126.623 \text{ MW}$$

$$P_{5New} = 147.9688 + (0.152)(150) = 173.7 \text{ MW}$$

$$P_{6New} = 57.8333 + (0.152)(150) = 85.2804 \text{ MW}$$

Setelah diperoleh besarnya daya pembangkitan masing-masing unit generator, fungsi biaya dari masing-masing unit pembangkitan adalah

Total biaya pembangkitan untuk semua unit adalah 15627.4036 \$/jam

Tabel 3.13 Hasil perhitungan ED menggunakan *base point and participation factor* dengan beban awal 1150 MW

Generator	$P_{iNew} \text{ (MW)}$	$F_i \text{ ($/jam)}$
1	448.515	4787.76085
2	172.59	2208.87642
3	293.289	3340.48066
4	126.623	1737.14733
5	173.7	2285.22825
6	85.2804	1267.91013
Total	1300	15627.4036

3.6.4 Kasus 2b

Permintaan beban awal pada kasus ini adalah 1250 MW dan permintaan beban baru tetap seperti 2a yaitu 1300 MW.

3.6.4.1 Dynamic Formulation Technique

Karena nilai permintaan beban baru pada kasus ini tetap seperti 2a, maka hasil perhitungan dengan dynamic formulation technique juga sama dengan Tabel 3.11.

3.6.4.2 Base Point and Participation Factor

$$\lambda_{min} = 8.4$$

$$\lambda_{max} = 14$$

$$\lambda_{start} = \frac{\lambda_{max} + \lambda_{min}}{2} = 11.2$$

$$\Delta\lambda = \frac{\lambda_{max} - \lambda_{min}}{2} = 2.8$$

Dengan memasukkan nilai lambda yang baru, kemudian melakukan iterasi pada lambda dengan cara menambah dan mengurangi nilai lambda sebesar $\frac{\Delta\lambda}{2}$. Hal ini bertujuan untuk mendapatkan hasil pembangkitan daya yang sesuai dengan load yang diinginkan. Dalam kasus ini permintaan beban awal yang diinginkan sebesar 1250 MW.

$$\lambda_{iter1} = 11.2 \dots \dots \dots totalgen = 690.9$$

$$\lambda_{iter2} = 14 \dots \dots \dots totalgen = 1470$$

$$\lambda_{iter3} = 12.6 \dots \dots \dots totalgen = 1062.5$$

$$\lambda_{iter4} = 13.3 \dots \dots \dots totalgen = 1307.6$$

$$\vdots$$

$$\lambda_{iter17} = 13.1420 \dots \dots \dots totalgen = 1250$$

Dengan permintaan beban awal 1250 diperoleh nilai λ 13.1420 \$/MWh, sehingga daya awal yang dibangkitkan masing-masing generator adalah

$$P_1 = 438.715 \text{ MW}$$

$$P_2 = 165.369 \text{ MW}$$

$$P_3 = 285.667 \text{ MW}$$

$$P_4 = 109.000 \text{ MW}$$

$$P_5 = 165.125 \text{ MW}$$

$$P_6 = 76.134 \text{ MW}$$

Participation Factor Untuk Masing-masing unit

$$\left(\frac{\Delta P_i}{\Delta P_D} \right) = \frac{\left(\frac{1}{F_i} \right)}{\sum_i \left(\frac{1}{F_i} \right)}$$

$$\left(\frac{\Delta P_1}{\Delta P_D} \right) = \frac{142.8571}{728.6759} = 0.196$$

$$\left(\frac{\Delta P_2}{\Delta P_D}\right) = \frac{105.2632}{728.6759} = 0.144$$

$$\left(\frac{\Delta P_3}{\Delta P_D}\right) = \frac{111.1111}{728.6759} = 0.152$$

$$\left(\frac{\Delta P_4}{\Delta P_D}\right) = \frac{111.1111}{728.6759} = 0.152$$

$$\left(\frac{\Delta P_5}{\Delta P_D}\right) = \frac{125}{728.6759} = 0.172$$

$$\left(\frac{\Delta P_6}{\Delta P_D}\right) = \frac{133.333}{728.6759} = 0.183$$

$$\Delta P_D = 1300 - 1250 = 50$$

$$P_{iNew} = P_{iBase} + \left(\frac{\Delta P_i}{\Delta P_D}\right) \Delta P_D$$

Daya baru yang dibangkitkan generator dari masing-masing unit pembangkit dan total biaya pembangkitannya bisa dilihat pada Tabel 3.14

Tabel 3.14 Hasil perhitungan ED menggunakan *base point and participation factor* dengan beban awal 1250 MW

Generator	$P_{iNew} (MW)$	$F_i (\$/jam)$
1	448.517	4787.78887
2	172.591	2208.89706
3	293.291	3340.50245
4	126.624	1737.16912
5	173.702	2285.25276
6	85.2823	1267.93628
Total	1300	15627.5465

BAB 4

HASIL SIMULASI DAN ANALISA

Pada Bab ini akan menampilkan hasil simulasi perhitungan ED dengan menggunakan *Dynamic Formulation Technique* dan *Base Point and Participation Factor*. Analisa yang dilakukan adalah membandingkan hasil perhitungan kedua metode tersebut. Diberikan beberap contoh kasus untuk menganalisa data yang dibutuhkan dalam Tugas Akhir.

4.1 Validasi Program *Dynamic Formulation Technique*

Program ED menggunakan *dynamic formulation technique* yang telah dibuat, di uji kebenarannya melalui uji validasi menggunakan data pembangkit yang telah tersedia pada referensi [5]. Apabila dalam proses simulasi diperoleh hasil yang sama dengan contoh pada referensi, maka program dikatakan telah dikembangkan dengan baik.

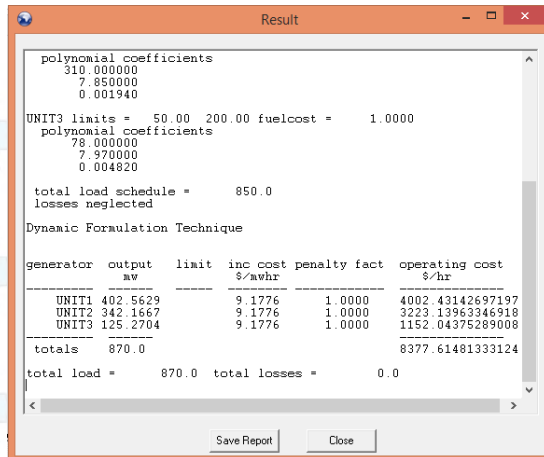
Data pembangkit yang digunakan sebagai masukan adalah sebagai berikut.

Tabel 4.1 Data kasus 1a

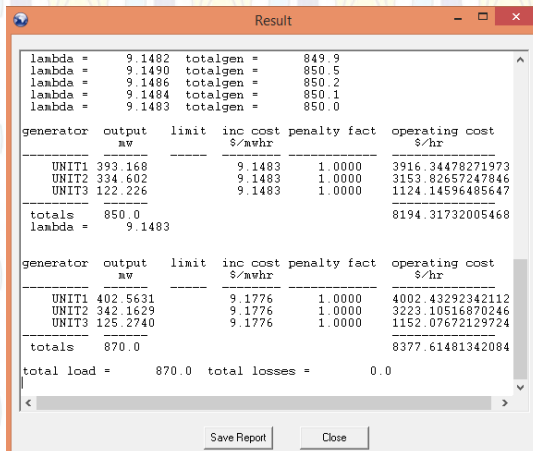
Unit	P_i^{min}	P_i^{max}	a_i	b_i	c_i
1	100	600	561	7.92	0.001562
2	100	450	310	7.85	0.00194
3	50	200	78	7.97	0.00482

4.1.1 Simulasi 1a

Permintaan beban awal untuk simulasi 1a adalah 850 MW. Kemudian ada permintaan penambahan beban sebesar 20 MW sehingga permintaan beban aktifnya menjadi 870 MW. Pada aplikasi Powergen yang digunakan, solution methods dipilih *dynamic formulation technique* dan *base point participation factor*. Dengan menjalankan program menggunakan, maka akan muncul tampilan seperti pada gambar 4.1 dan 4.2.



Gambar 4.1 Tampilan hasil perhitungan *dynamic formulation technique* kasus 1a



Gambar 4.2 Tampilan hasil perhitungan *base point and participation factor* kasus 1a

Metode yang digunakan sebagai pembanding dari *dynamic formulation technique* selain metode *base point and participation factor* adalah metode *quadratic programming*.

Adapun hasil perbandingan perhitungan ketiga metode tersebut dapat dilihat pada tabel 4.2 dan 4.3.

Tabel 4.2 Hasil perhitungan kasus 1a

Unit	850 MW ke 870 MW			
	Base Point and Participation Factor		Dynamic Formulation Technique	
	P_{iNew} (MW)	F_i \$/jam	P_{iNew} (MW)	F_i \$/jam
1	402.5631	4002.4329	402.5629	4002.4314
2	342.1669	3223.1051	342.1667	3223.1396
3	125.2700	1152.0767	125.2704	1152.0437
Tot	870	8377.6147	870	8377.6147

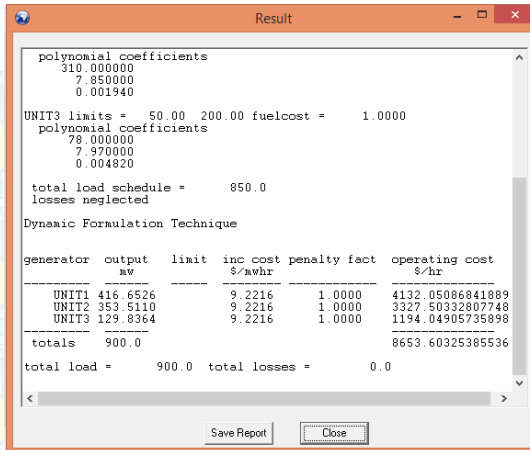
Tabel 4.3 Lanjutan hasil perhitungan kasus 1a

Unit	870 MW	
	Quadratic Programming	
	P_{iNew} (MW)	F_i \$/jam
1	402.5629	4002.4
2	342.1667	3223.1
3	125.2704	1152.0
Tot	870	8377.6

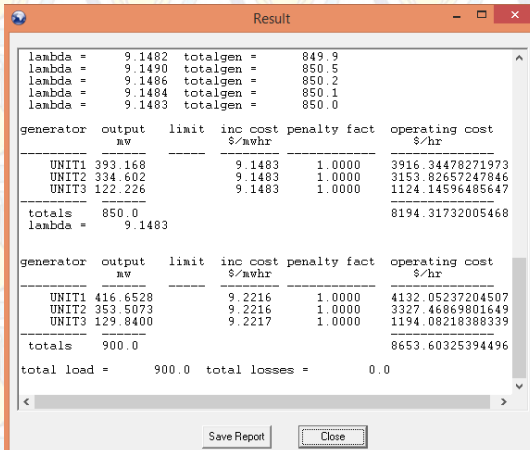
4.1.2 Simulasi 1b

Data yang dimasukkan untuk simulasi 1b sama dengan simulasi 1a yaitu dari referensi [5], namun pada simulasi 1b besarnya penambahan beban adalah 50 MW sehingga permintaan beban aktif yang baru menjadi 900 MW sedangkan permintaan beban awal tetap diisi 850 MW.

Hasil dari simulasi untuk kasus 1b dengan memilih metode *dynamic formulation technique* dan *base point and participation factor* pada *software* Powergen dapat dilihat pada gambar 4.4 dan 4.5.



Gambar 4.3 Tampilan hasil perhitungan *dynamic formulation technique* kasus 1b



Gambar 4.4 Tampilan hasil perhitungan *base point and participation factor* kasus 1b

Adapun hasil perbandingan perhitungan ketiga metode tersebut dapat dilihat pada tabel 4.4 dan 4.5.

Tabel 4.4 Hasil perhitungan kasus 1b

Unit	850 MW ke 900 MW			
	Base Point and Participation Factor		Dynamic Formulation Technique	
	P_{iNew} (MW)	F_i (\$/jam)	P_{iNew} (MW)	F_i (\$/jam)
1	416.6528	4132.0523	416.6526	4132.0509
2	353.5113	3327.4687	353.5110	3327.5033
3	129.8360	1194.0822	129.8364	1194.0491
Tot	900	8653.6032	900	8653.6032

Tabel 4.5 Lanjutan hasil perhitungan kasus 1b

Unit	900 MW	
	Quadratic Programming	
	P_{iNew} (MW)	F_i \$/jam
1	416.6526	4132.1
2	353.511	3327.5
3	129.8364	1194.0
Tot	870	8653.6

4.1.3 Analisa

Jika diperhatikan, nilai pembangkitan hasil simulasi dengan menggunakan *software* powergen yang telah dikembangkan untuk unit 1 hingga unit 3 sudah sama dengan hasil pada perhitungan manual (Subsub Bab 3.2.1) maupun hasil pada referensi [5]. Selain itu hasil perhitungan yang dihasilkan juga sama dengan hasil perhitungan dengan menggunakan metode *quadratic programming*. Sehingga dapat disimpulkan bahwa program ED metode *dynamic formulation technique* yang dikembangkan ini telah dapat digunakan untuk memecahkan permasalahan ED.

Hasil penyelesaian ED menggunakan *dynamic formulation technique* dan *base point and participation factor* menghasilkan biaya pembangkitan yang sama.

4.2 Sistem Dengan 6 Unit Pembangkit

Setelah dilakukan uji validasi menggunakan sistem dengan 3 unit pembangkit, program ED digunakan untuk melakukan perhitungan pada

kasus 2. Kasus 2 merupakan kasus sistem dengan 6 unit pembangkit. Data pembangkit diambil dari referensi [6] yaitu seperti pada tabel 4.6.

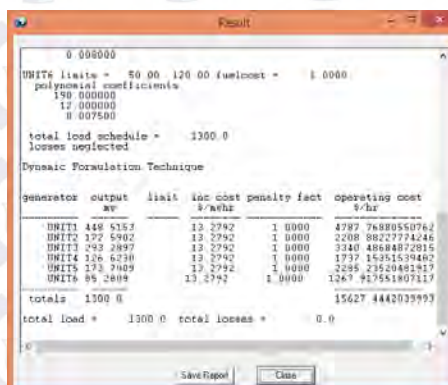
Perhitungan untuk kasus 2 akan dipecah menjadi 2 bagian yaitu perhitungan kasus 2a dan perhitungan kasus 2b. Pada kasus 2a, nilai permintaan beban awal adalah 1150 MW dan permintaan beban baru adalah 1300 MW. Sedangkan untuk kasus 2b, nilai permintaan beban awal adalah 1250 MW sedangkan nilai permintaan beban baru sama dengan pada kasus 2a yaitu 1300 MW.

Tabel 4.6 Data kasus 2

Generator	P_i^{max}	P_i^{min}	a_i	b_i	c_i
1	500	100	240	7	0.007
2	200	50	200	10	0.0095
3	300	80	220	8	0.009
4	150	50	200	11	0.009
5	200	50	220	10.5	0.008
6	120	50	190	12	0.0075

4.2.1 Simulasi 2a

Hasil dari simulasi untuk kasus 2a dapat dilihat pada gambar 4.5 dan 4.6.



```

0 000000
UNIT6 limits = 50.00 120.00 fuelcost = 1.0000
polynomial coefficients
150 000000
12 000000
0 007500

total load schedule = 1300.0
losses neglected

Dynamic Formulation Technique

generator output limit inc cost penalty fact operating cost
          mw          $/mwh
UNIT1 448 6153      13.2792      1.0000      4787 76880550762
UNIT2 152 6462      13.2792      1.0000      2208 88227774246
UNIT3 283 2887      13.2792      1.0000      3340 48684872815
UNIT4 126 6230      13.2792      1.0000      1737 15351539462
UNIT5 173 7409      13.2792      1.0000      2285 23520481819
UNIT6 85 2809      13.2792      1.0000      1267 817551907117

totals 1300 0      15627 4442035993
total load = 1300.0 total losses = 0.0
  
```

Gambar 4.5 Tampilan hasil perhitungan *dynamic formulation technique* kasus 2a

Result						
generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr	
UNIT1	419.107	12.8675	1.0000		4403.30445840023	
UNIT2	150.921	12.8675	1.0000		1925.59275882122	
UNIT3	270.417	12.8675	1.0000		3041.45902320018	
UNIT4	103.750	12.8675	1.0000		1438.12568986684	
UNIT5	147.969	12.8675	1.0000		1948.82890110020	
UNIT6	57.833	12.8675	1.0000		909.084161173551	
totals					13666.3949925622	
lambda =					12.8675	
generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr	
UNIT1	448.5175	13.2792	1.0000		4787.79879405768	
UNIT2	172.5887	13.2792	1.0000		2208.86244020346	
UNIT3	293.2925	13.2793	1.0000		3340.52492788452	
UNIT4	126.6225	13.2792	1.0000		1737.14733043220	
UNIT5	173.7016	13.2792	1.0000		2285.24484575585	
UNIT6	85.2770	13.2792	1.0000		1267.865865916329	
totals					15627.4442042500	
total load =			1300.0	total losses =		0.0

Gambar 4.6 Tampilan hasil perhitungan *base point and participation factor* kasus 2a

Pada simulasi program, digunakan 10 angka dibelakang koma. Hal tersebut mengacu pada referensi [5]. Namun jumlah angka dibelakang koma yang ditampilkan pada tabel hanya 4 angka saja. Hasil simulasi program dari *dynamic formulation technique*, *base point and participation factor* dan *quadratic programming* jika dibuat dalam tabel, hasilnya dapat dilihat pada tabel 4.7 dan 4.8.

Tabel 4. 7 Hasil perhitungan kasus 2a

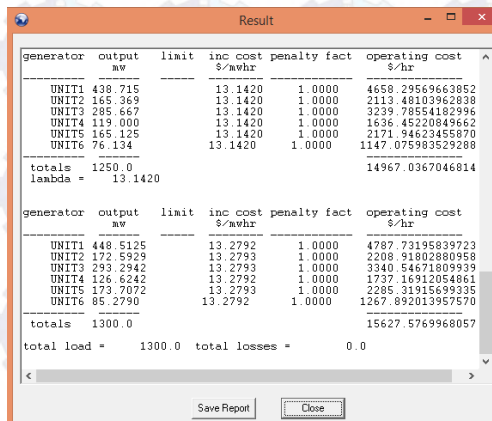
Unit	1150 MW ke 1300 MW			
	Base Point and Participation Factor		Dynamic Formulation Technique	
	P_{iNew} (MW)	F_i (\$/jam)	P_{iNew} (MW)	F_i (\$/jam)
1	448.5175	4787.7988	448.5153	4787.7688
2	172.5887	2208.8624	172.5902	2208.8823
3	293.2925	3340.5249	293.2897	3340.4868
4	126.6225	1737.1473	126.623	1737.1535
5	173.7016	2285.2448	173.7009	2285.2352
6	85.2770	1267.8659	85.2809	1267.9175
Tot	1300	15627.4441	1300	15627.4441

Tabel 4.8 Lanjutan hasil perhitungan kasus 2a

Unit	1300 MW	
	Quadratic Programming	
	P_{iNew} (MW)	F_i (\$/jam)
1	448.5153	4787.8
2	172.5902	2208.9
3	293.2897	3340.5
4	126.623	1737.2
5	173.7009	2285.2
6	85.2809	1267.9
Tot	1300	15627

4.2.2 Simulasi 2b

Untuk kasus 2b, nilai permintaan beban awal diisi 1250 MW sedangkan nilai permintaan beban baru sama dengan pada kasus 2a yaitu 1300 MW.



generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr
UNIT1	438.715	13.1420	1.0000		4658.29569663852
UNIT2	165.369	13.1420	1.0000		2113.48103962838
UNIT3	285.667	13.1420	1.0000		3239.7854182996
UNIT4	119.000	13.1420	1.0000		1636.45220849662
UNIT5	165.125	13.1420	1.0000		2171.94623455870
UNIT6	76.134	13.1420	1.0000		1147.075983529288
totals	1250.0				14967.0367046814
lambda = 13.1420					

generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr
UNIT1	448.5125	13.2792	1.0000		4787.73195839723
UNIT2	172.5929	13.2793	1.0000		2208.91002880958
UNIT3	293.2942	13.2793	1.0000		3340.54671809939
UNIT4	126.6242	13.2792	1.0000		1737.16912054861
UNIT5	173.7072	13.2793	1.0000		2285.31915699335
UNIT6	85.2790	13.2792	1.0000		1267.092013957570
totals	1300.0				15627.5769968057
total load = 1300.0 total losses = 0.0					

Gambar 4.7 Tampilan hasil perhitungan *base point* and *participation factor* kasus 2b

Tabel 4.9 Hasil perhitungan kasus 2b

Unit	1250 MW ke 1300 MW			
	Base Point and Participation Factor		Dynamic Formulation Technique	
	P_{iNew} (MW)	F_i (\$/jam)	P_{iNew} (MW)	F_i (\$/jam)
1	448.5125	4787.7319	448.5153	4787.7688
2	172.5929	2208.9180	172.5902	2208.8823
3	293.2942	3340.5467	293.2897	3340.4868
4	126.6242	1737.1691	126.623	1737.1535
5	173.7072	2285.3191	173.7009	2285.2352
6	85.279	1267.8920	85.2809	1267.9175
Tot	1300	15627.5768	1300	15627.4441

4.2.3 Analisa

Jika pada kasus 1a dan 1b menggunakan nilai permintaan beban awal yang sama namun permintaan beban baru yang berbeda, pada kasus 2a dan 2b menggunakan nilai permintaan beban awal berbeda namun nilai permintaan beban baru sama. Dari perhitungan ED menggunakan *dynamic formulation technique* dan perhitungan ED menggunakan *base point and participation factor* menghasilkan perhitungan yang sedikit berbeda.

Dari tabel 4.6 dan 4.8 dapat dilihat bahwa jika nilai permintaan beban awal berbeda, maka biaya pembangkitan pada hasil perhitungan menggunakan *base point and participation factor* juga berbeda walaupun permintaan beban baru memiliki nilai yang sama. Hal ini disebabkan karena metode *base point and participation factor* membutuhkan perhitungan penyelesaian ED terlebih dahulu. Perhitungan tersebut kemudian digunakan sebagai daya awal perhitungan untuk permintaan beban yang baru. Sedangkan untuk *dynamic formulation technique* tidak memerlukan inisialisasi daya awal, sehingga berapapun permintaan beban awal tidak akan mempengaruhi perhitungan ED untuk permintaan beban yang baru.

Jika dibandingkan, antara perhitungan ED menggunakan *dynamic formulation technique* dan *base point and participation factor* terdapat selisih biaya pembangkitan pada kasus 2b. Tidak terdapat selisih biaya pembangkitan untuk kasus 2a. Sedangkan selisih biaya pembangkitan

untuk kasus 2b adalah 0.13279 \$/jam. Penyelesaian ED menggunakan *dynamic formulation technique* menghasilkan biaya pembangkitan yang lebih rendah bila dibandingkan dengan *base point and participation factor*.

4.3 Sistem Dengan 18 Unit Pembangkit

Kasus selanjutnya adalah kasus 3 yaitu sistem dengan 18 unit pembangkit. Data pembangkit diambil dari referensi [7] yaitu seperti pada tabel 4.10.

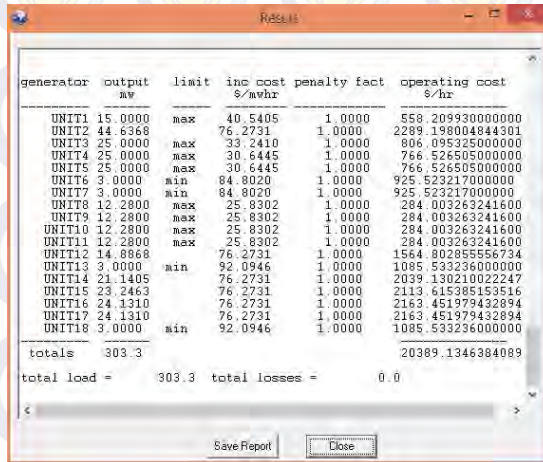
Kasus ini akan dibagi lagi menjadi 2 bagian, yang pertama adalah nilai permintaan beban awal 250 MW dan ada penambahan beban sebesar 53.3 MW sehingga permintaan beban baru menjadi 303.3 MW. Sedangkan yang kedua adalah nilai permintaan beban awal 275 MW dan ada penambahan beban sebesar 28.3 MW sehingga permintaan beban baru menjadi 303.3 MW.

Tabel 4.10 Data kasus 3

Unit	P_i^{min}	P_i^{max}	a_i	b_i	c_i
1	7	15.00	85.74158	22.45526	0.602842
2	7	45.00	85.74158	22.45526	0.602842
3	13	25.00	108.9837	22.52789	0.214263
4	16	25.00	49.06263	26.75263	0.077837
5	16	25.00	49.06263	26.75263	0.077837
6	3	14.75	677.73000	80.39345	0.734763
7	3	14.75	677.73000	80.39345	0.734763
8	3	12.28	44.39000	13.19474	0.514474
9	3	12.28	44.39000	13.19474	0.514474
10	3	12.28	44.39000	13.19474	0.514474
11	3	12.28	44.39000	13.19474	0.514474
12	3	24.00	574.96030	56.70947	0.657079
13	3	16.20	820.37760	84.67579	1.236474
14	3	36.20	603.02370	59.59026	0.394571
15	3	45.00	567.93630	56.70947	0.420789
16	3	37.00	567.93630	55.96500	0.420789
17	3	45.00	567.93630	55.96500	0.420789
18	3	16.20	820.37760	84.67579	1.236474

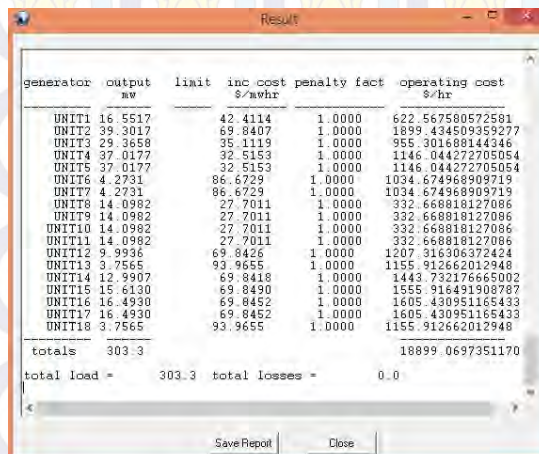
4.3.1 Simulasi 3a

Hasil dari simulasi untuk kasus 3a dengan memilih *dynamic formulation technique* dan *base point and participation factor* dapat dilihat pada gambar 4.7 dan 4.8.



generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr
UNIT1	15.0000	max	40.5405	1.0000	558.2099300000000
UNIT2	44.6368		76.2731	1.0000	2289.198004844301
UNIT3	25.0000	max	33.2410	1.0000	806.0953250000000
UNIT4	25.0000	max	30.6445	1.0000	766.5265050000000
UNIT5	25.0000	max	30.6445	1.0000	766.5265050000000
UNIT6	3.0000	min	84.8020	1.0000	925.5232170000000
UNIT7	3.0000	min	84.8020	1.0000	925.5232170000000
UNIT8	12.2800	max	25.8302	1.0000	284.003263241600
UNIT9	12.2800	max	25.8302	1.0000	284.003263241600
UNIT10	12.2800	max	25.8302	1.0000	284.003263241600
UNIT11	12.2800	max	25.8302	1.0000	284.003263241600
UNIT12	14.8868		76.2731	1.0000	1564.802855556734
UNIT13	3.0000	min	92.0946	1.0000	1085.5332360000000
UNIT14	21.1405		76.2731	1.0000	2039.130210022247
UNIT15	23.2463		76.2731	1.0000	2113.615385153516
UNIT16	24.1310		76.2731	1.0000	2163.451979432894
UNIT17	24.1310		76.2731	1.0000	2163.451979432894
UNIT18	3.0000	min	92.0946	1.0000	1085.5332360000000
totals	303.3				20389.1346384089
total load = 303.3 total losses = 0.0					

Gambar 4.7 Tampilan hasil perhitungan *dynamic formulation technique* kasus 3



generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr
UNIT1	16.5517		42.4114	1.0000	622.567580572581
UNIT2	39.3017		69.8407	1.0000	1899.434509359277
UNIT3	29.3658		35.1119	1.0000	955.30168044346
UNIT4	37.0177		32.5153	1.0000	1146.044272705054
UNIT5	37.0177		32.5153	1.0000	1146.044272705054
UNIT6	4.2731		86.6729	1.0000	1034.674968909719
UNIT7	4.2731		86.6729	1.0000	1034.674968909719
UNIT8	14.0982		27.7011	1.0000	332.668818127086
UNIT9	14.0982		27.7011	1.0000	332.668818127086
UNIT10	14.0982		27.7011	1.0000	332.668818127086
UNIT11	14.0982		27.7011	1.0000	332.668818127086
UNIT12	9.9936		69.8426	1.0000	1207.316306372424
UNIT13	3.7565		93.9655	1.0000	1155.912662012948
UNIT14	12.9907		69.8418	1.0000	1443.732176665002
UNIT15	15.6130		69.8490	1.0000	1555.916491908787
UNIT16	16.4930		69.8452	1.0000	1605.430951165433
UNIT17	16.4930		69.8452	1.0000	1605.430951165433
UNIT18	3.7565		93.9655	1.0000	1155.912662012948
totals	303.3				18899.0697351170
total load = 303.3 total losses = 0.0					

Gambar 4.8 Tampilan hasil perhitungan *base point and participation factor* kasus 3a tanpa batas pembangkitan generator.

generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr
UNIT1	15.0000	max	40.5405	1.0000	558.209930000000
UNIT2	42.8952	max	74.1733	1.0000	2158.18950516011
UNIT3	25.0000	max	33.2410	1.0000	806.095325000000
UNIT4	25.0000	max	30.6445	1.0000	766.526505000000
UNIT5	25.0000	max	30.6445	1.0000	766.526505000000
UNIT6	7.2214	91.0055	1.0055	1.0000	1296.598781550086
UNIT7	7.2214	91.0055	1.0055	1.0000	1296.598781550086
UNIT8	12.2800	max	25.8302	1.0000	284.003263241600
UNIT9	12.2800	max	25.8302	1.0000	284.003263241600
UNIT10	12.2800	max	25.8302	1.0000	284.003263241600
UNIT11	12.2800	max	25.8302	1.0000	284.003263241600
UNIT12	13.2905	74.1752	1.0000	1.0000	1444.719542136849
UNIT13	5.5085	98.2981	1.0000	1.0000	1324.335021910398
UNIT14	18.4810	74.1744	1.0000	1.0000	1839.075033489785
UNIT15	20.7612	74.1816	1.0000	1.0000	1976.664146897820
UNIT16	21.6412	74.1778	1.0000	1.0000	1976.158624229169
UNIT17	21.6412	74.1778	1.0000	1.0000	1976.158624229169
UNIT18	5.5085	98.2981	1.0000	1.0000	1324.335021910398
totals	303.3				20596.2044463861
total load =	303.3				total losses = 0.0

Gambar 4.9 Tampilan hasil perhitungan *base point and participation factor* kasus 3a dengan batas pembangkitan generator

Tabel 4.11 Hasil perhitungan kasus 3a

Unit	250 MW ke 303.3 MW			
	Base Point and Participation Factor		Dynamic Formulation Technique	
	P_{iNew} (MW)	F_i (\$/jam)	P_{iNew} (MW)	F_i (\$/jam)
1	15	558.20993	15	558.2099
2	42.8952	21581895	44.6368	2289.198
3	25	806.0953	25	806.0953
4	25	766.5265	25	766.5265
5	25	766.5265	25	766.5265
6	7.2214	1296.5988	3	925.5232
7	7.2214	1296.5988	3	925.5232
8	12.28	284.0032	12.28	284.0032
9	12.28	284.0032	12.28	284.0032
10	12.28	284.0032	12.28	284.0032
11	12.28	284.0032	12.28	284.0032
12	13.2905	1444.7195	14.8868	1564.803
13	5.5085	1324.335	3	1085.5332
14	18.481	1839.075	21.1405	2039.1302
15	20.7612	1926.6641	23.2463	2113.6154

Tabel 4.12 Lanjutan hasil perhitungan kasus 3a

Unit	250 MW ke 303.3 MW			
	Base Point and Participation Factor		Dynamic Formulation Technique	
	P_{iNew} (MW)	F_i (\$/jam)	P_{iNew} (MW)	F_i (\$/jam)
16	21.6412	1976.1586	24.131	2163.452
17	21.6412	1976.1586	24.131	2163.452
18	5.5085	1324.335	3	1085.5332
Tot	303.3	20596.2044	303.3	20389.1346

Tabel 4.13 Lanjutan hasil perhitungan kasus 3a

Unit	303.3 MW	
	Quadratic Programming	
	P (MW)	F \$/jam
1	15	558.2
2	44.6368	2289.2
3	25	806.1
4	25	766.5
5	25	766.5
6	3	925.5
7	3	925.5
8	12.28	284
9	12.28	284
10	12.28	284
11	12.28	284
12	14.8868	1564.8
13	3	1085.5
14	21.1405	2039.1
15	23.2463	2113.6
16	24.131	2163.4
17	24.131	2163.4
18	3	1085.5
Total	303.3	20389.1

4.3.2 Simulasi 3b

Hasil dari simulasi untuk kasus 3b dengan memilih *base point* and *participation factor* dapat dilihat pada gambar 4.9.

Result

generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr
UNIT1	15.8239	41.5339	1.0000	592.019586037680	
UNIT2	41.8039	72.8575	1.0000	2077.963670617032	
UNIT3	27.3180	34.2344	1.0000	884.300421376126	
UNIT4	31.3809	31.6378	1.0000	965.234431711113	
UNIT5	31.3809	31.6378	1.0000	965.234431711113	
UNIT6	3.6760	85.7954	1.0000	983.181508632006	
UNIT7	3.6760	85.7954	1.0000	983.181508632006	
UNIT8	13.2454	26.8236	1.0000	309.419004569972	
UNIT9	13.2454	26.8236	1.0000	309.419004569972	
UNIT10	13.2454	26.8236	1.0000	309.419004569972	
UNIT11	13.2454	26.8236	1.0000	309.419004569972	
UNIT12	12.2859	72.8550	1.0000	1370.866938287219	
UNIT13	3.4017	93.0880	1.0000	1122.725435253000	
UNIT14	16.8188	72.8626	1.0000	1716.870247086472	
UNIT15	19.1903	72.8596	1.0000	1811.172967523381	
UNIT16	20.0703	72.8557	1.0000	1860.673531958063	
UNIT17	20.0703	72.8557	1.0000	1860.673531958063	
UNIT18	3.4017	93.0880	1.0000	1122.725435253000	
totals	303.3				19554.4996544361
total load =	303.3	total losses =	0.0		

Save Report

Close

Gambar 4.10 Tampilan hasil perhitungan *base point* and *participation factor* kasus 3b tanpa batas generator



generator	output mw	limit	inc cost \$/mwhr	penalty fact	operating cost \$/hr
UNIT1	15.0000	max	40.5405	1.0000	558.2099300000000
UNIT2	43.7113	max	75.1579	1.0000	2219.168851974248
UNIT3	25.0000	max	33.2410	1.0000	806.0953250000000
UNIT4	25.0000	max	30.6445	1.0000	766.5265050000000
UNIT5	25.0000	max	30.6445	1.0000	766.5265050000000
UNIT6	5.2414	max	88.0958	1.0000	1119.287490883610
UNIT7	5.2414	max	88.0958	1.0000	1119.287490883610
UNIT8	12.2800	max	25.8302	1.0000	284.003263241600
UNIT9	12.2800	max	25.8302	1.0000	284.003263241600
UNIT10	12.2800	max	25.8302	1.0000	284.003263241600
UNIT11	12.2800	max	25.8302	1.0000	284.003263241600
UNIT12	14.0364	max	75.1553	1.0000	1500.412340062489
UNIT13	4.3319	max	95.3884	1.0000	1210.388957208425
UNIT14	19.7338	max	75.1631	1.0000	1932.624324585215
UNIT15	21.9238	max	75.1600	1.0000	2013.475842719799
UNIT16	22.8038	max	75.1562	1.0000	2062.965807614407
UNIT17	22.8038	max	75.1562	1.0000	2062.965807614407
UNIT18	4.3319	max	95.3884	1.0000	1210.388957208425
totals	303.3				20484.3371886710
total load =		303.3	total losses =		0.0

Gambar 4.11 Tampilan hasil perhitungan *base point* and *participation factor* kasus 3b dengan batas generator

Tabel 4.14 Hasil perhitungan kasus 3b

Unit	303.3 MW			
	Base Point and Participation Factor		Dynamic Formulation Technique	
	P_{iNew} (MW)	F_i (\$/jam)	P_{iNew} (MW)	F_i (\$/jam)
1	15	558.20993	15	558.2099
2	43.7119	2219.169	44.6368	2289.198
3	25	806.0953	25	806.0953
4	25	766.5265	25	766.5265
5	25	766.5265	25	766.5265
6	5.2414	1119.2875	3	925.5232
7	5.2414	1119.2875	3	925.5232
8	12.28	284.0032	12.28	284.0032
9	12.28	284.0032	12.28	284.0032
10	12.28	284.0032	12.28	284.0032
11	12.28	284.0032	12.28	284.0032
12	14.0364	1500.4123	14.8868	1564.803
13	4.3319	1210.389	3	1085.5332
14	19.7338	1932.6243	21.1405	2039.1302
15	21.9238	2013.4758	23.2463	2113.6154
16	22.8038	2062.9658	24.131	2163.452
17	22.8038	2062.9658	24.131	2163.452
18	4.3319	1210.389	3	1085.5332
Tot	303.3	20484.3372	303.3	20389.1346

4.3.3 Analisa

Pada kasus 3a dan 3b menggunakan nilai permintaan beban awal berbeda namun pada menggunakan nilai permintaan beban baru sama. Dari perhitungan ED menggunakan *dynamic formulation technique* dan perhitungan ED menggunakan *base point and participation factor* menghasilkan perhitungan yang sedikit berbeda.

Dari tabel 4.12 dan 4.14 dapat dilihat bahwa jika nilai permintaan beban awal berbeda, maka biaya pembangkitan pada hasil perhitungan menggunakan *base point and participation factor* juga berbeda walaupun permintaan beban baru memiliki nilai yang sama. Hal ini disebabkan karena metode *base point and participation factor* membutuhkan

perhitungan penyelesaian ED terlebih dahulu. Perhitungan tersebut kemudian digunakan sebagai daya awal perhitungan untuk permintaan beban yang baru. Sedangkan untuk *dynamic formulation technique* tidak memerlukan inisialisasi daya awal, sehingga berapapun permintaan beban awal tidak akan mempengaruhi perhitungan ED untuk permintaan beban yang baru.

Untuk kasus 3, pada perhitungan menggunakan *base point and participation factor* terdapat 2 macam perhitungan, perhitungan pertama adalah ketika dianggap tidak terdapat batas minimum dan maksimum pembangkitan pada masing-masing unit pembangkit dan perhitungan kedua adalah ketika pada unit pembangkitan terdapat daya minimum dan maksimum yang harus dibangkitkan.

Dilakukan 2 macam perhitungan karena pada kasus 3 ketika perhitungan menentukan daya awal yang optimal untuk masing-masing generator menggunakan iterasi lambda, hasil perhitungan menunjukkan bahwa terdapat beberapa unit pembangkit yang membangkitkan daya pada batas minimum atau maksimum kemampuan generator. Sehingga untuk perhitungan pertama, pada bagian partisipasi factor, generator dianggap tidak memiliki daya minimum atau maksimum yang harus dibangkitkan. Saat daya yang dibangkitkan suatu generator bernilai maksimum, maka pada bagian partisipasi factor, unit tersebut akan membangkitkan daya yang lebih besar lagi. Hal tersebut bisa dilihat pada gambar 4.8 dan 4.10.

Untuk perhitungan yang kedua adalah terdapat batas daya pembangkitan untuk masing-masing unit pembangkit, sehingga setelah didapat daya awal pembangkitan pada masing-masing unit pembangkit, unit pembangkit yang telah mencapai batas maksimal tidak akan diikutkan dalam perhitungan untuk factor partisipasi. Pembagian faktor partisipasi hanya untuk unit pembangkitan yang nilainya masih di bawah batas maksimal pembangkitan generator.

Contoh bisa dilihat pada gambar 4.11. Daya yang dibangkitkan pada unit 1, 3, 4, 5, 8, 9, 10, 11 setelah dilakukan optimasi menggunakan iterasi lambda telah mencapai batas maksimal daya yang bisa dibangkitkan generator. Oleh karena itu, unit-unit tersebut tidak diperhitungkan pada factor partisipasi. Pembagian faktor partisipasi hanya dilakukan pada unit 2, 6, 7, 12, 13, 14, 15, 16, 17, 18. Hasil dari perhitungan *base point and participation factor* dengan memperhatikan batas pembangkitan generator tersebut bisa dilihat pada gambar 4.11 maupun pada tabel 4.10 dan 4.12. Terlihat biaya pembangkitan yang diperoleh dengan

menggunakan *dynamic formulation technique* lebih minimum jika dibandingkan dengan *base point and participation factor*. Terdapat selisih biaya 207.0698 \$/jam pada kasus 3a dan 95. 2026 \$/jam pada kasus 3b karena pendistribusian daya pada metode *base point and participation factor* sangat tergantung pada besarnya daya pembangkitan awal masing-masing unit pembangkit, sehingga pada metode *base point and participation factor*, daya unit dengan *incremental cost* yang mahal tetap akan bertambah sesuai dengan pembagian *participation factor*. Berbeda dengan metode *dynamic formulation technique*, daya pembangkitan unit dengan *incremental cost* yang mahal akan diset minimum dan pendistribusian daya akan dibagi kepada unit yang mempunyai *incremental cost* lebih rendah agar biaya pembangkitan yang dihasilkan lebih rendah.



BAB 5

PENUTUP

5.1 Kesimpulan

Dari semua proses yang meliputi studi literatur, serta simulasi dan analisis, maka terdapat beberapa hal yang dapat disimpulkan terkait Tugas Akhir ini, yaitu:


1. Aplikasi perhitungan ED menggunakan *dynamic formulation technique* tanpa memperhitungkan rugi transmisi dapat melakukan perhitungan sesuai dengan batasan daya yang diijinkan untuk masing-masing unit pembangkit.
2. Jika dibandingkan dengan metode *base point and participation factor*, biaya pembangkitan yang dihasilkan dari perhitungan menggunakan *dynamic formulation technique* lebih minimum yaitu untuk kasus 2b sebesar 0.13279 \$/jam. Sedangkan pada kasus 3, selisih biaya pembangkitan sebesar 207.0698 \$/jam pada kasus 3a dan 95. 2026 \$/jam pada kasus 3b
3. Perhitungan ED menggunakan *dynamic formulation technique* tidak memerlukan daya pembangkitan awal untuk inialisasi sehingga permintaan beban awal tidak mempengaruhi perhitungan ketika ada penambahan permintaan beban.
4. Perhitungan ED menggunakan *base point and participation factor* dipengaruhi oleh besarnya perubahan permintaan beban. Sehingga untuk 2 kondisi, walaupun dengan permintaan beban baru yang sama jika permintaan beban awal berbeda maka ada kemungkinan biaya pembangkitan yang dihasilkan berbeda.
5. Hasil akhir dari aplikasi perhitungan yang dikembangkan dapat digunakan untuk meng-*upgrade* aplikasi perhitungan pada *software* powergen yang telah digunakan sebelumnya.

5.2 Saran

Adapun saran untuk penelitian selanjutnya yang berkaitan dengan Tugas Akhir ini, yaitu:

1. Aplikasi perhitungan ED dapat dikembangkan dengan memberikan perhitungan dengan memperhitungkan rugi-rugi transmisi.
2. Aplikasi ED dapat dikembangkan dengan menampilkan gambar yang lebih menarik.

DAFTAR PUSTAKA

- 
- [1] Penangsang, O., “Analisis Aliran Daya”, ITS Press Surabaya, 2012
 - [2] Wibowo, R. S., Nursidi, Satriyadi H, I. G. N., Uman P, D. F., Soeprijanto, A., Penangsang, O., “Dynamic DC Optimal Power Flow using Quadratic Programming”, International Conference on Information Technology and Electrical Engineering (ICITEE), 360-364,2013
 - [3] Hadi S., “Power System Analysis 2nd Edition”,McGrowHill, Ch1,1999.
 - [4] Wood, A. J. dan Wollenberg, B. F., ”Power Generation, Operation and Control”, Wiley., New York, 3rd ed., 2013.
 - [5] I. Mohammed, Abouheaf, Lee Wei-Jen, L. Lewis Frank: ‘Dynamic formulation and approximation methods to solve economic dispatch problems’, IET Generation, Transmission and Distribution, 2013.
 - [6] F. Benhamida, et all “Constrained Dynamic Economical Dispatch using a Compact Quadratic Programming Method including Losses” IEEE 2013.
 - [7] F. Benhamida, et all “Solving Dynamic Economic Load Dispatch With Ramp Rate Limit Using Quadratic Programming” IEEE 2013.
 - [8] Borland Delphi 7, “Developer’s Guide”, Borland Software Corporation, 2002



LAMPIRAN

```
procedure prod_cost(    i : integer;
                        unitmw : real;
                        var unitcost : real );

    { Routine to return unit production cost given unit output in mw }
    { input : unit index = i }
    {    unit MW = unitmw }
    { output: unit production cost = unitcost }

var
    j : integer;
    partmw, unitihr, segmentcost : real;

label return;

begin

case curvetype of
    poly :              {Polynomial I/O curve}
        begin
            unitcost := 0;
            for j := curveorder downto 1 do
                unitcost := ( unitcost + coeff[ i,j ] ) * unitmw;
            unitcost := unitcost + coeff[ i,0 ];
            unitcost := unitcost * fuelcost[ i ];
            goto return
        end;
    end;

return:

end; { End procedure }
```

```

procedure output_routine( var outfile : text;
                        lambda : real );

var
    limittxt : string[5];
    totalgen, totalcost, totalload, totnewload : real;
    unitihr, unitinccost, unitcost, deltap2 : real;
    i : integer;

label return;

begin
    writeln(outfile);
    writeln(outfile,
        'generator output limit inc cost penalty fact operating cost');
    writeln(outfile,
        '      mw      $/mwhr      $/hr      ');
    writeln(outfile,
        '-----');

    totalgen := 0.0;
    totalcost := 0.0;

    for i := 1 to ngen do
        begin
            write(outfile, genname[i]:9);
            write(outfile, ', p[i]:6:4, ' ');
            limittxt := ' ';
            if abs( p[i] - pmin[i] ) < total_gen_tolerance then limittxt := 'min ';
            if abs( p[i] - pmax[i] ) < total_gen_tolerance then limittxt := 'max ';
            write(outfile, limittxt );

            ihr_ftn( i, p[ i ], unitihr ) ; {Get unit incremental heat rate}

            unitinccost := unitihr * fuelcost[i];
            write(outfile, unitinccost:9:4);
            write(outfile, ' ', penfac[i]:9:4, ' ');
        end
    end;

return;
end;

```

```

prod_cost( i, p[ i ], unitcost );      {Calculate unit operating cost}

writeln(outfile, '',unitcost:9:12);
totalgen := totalgen + p[i];
//deltap2 := totalgen;

totalcost := totalcost + unitcost;
end;
writeln(outfile,
'-----' );
write(outfile, ' totals');
write(outfile, totalgen:9:1,
', totalcost:9:12);

writeln(outfile);
//writeln(outfile, ' lambda = ', lambda:10:4 );
writeln(outfile);

if (schedtype = totgen ) and ( losstype <> lossform ) then goto return;

if schedtype=totload then
begin
totalload := schedmw;
totnewload := newload;
end ;

if schedtype=totgen then totalload := totalgen - mwlosses;

if (solution_type = lamsearch) or (solution_type = tbllookup) then
begin
writeln(outfile, 'total load = ',totalload:10:1,
' total losses = ',mwlosses:10:1);
end
else
begin
writeln(outfile, 'total load = ',totnewload:10:1,
' total losses = ',mwlosses:10:1);
end;

```


return:

end; { End procedure }

```
procedure order_routine(      numorder : integer;
                             ordertable : system_ihr_array_real;
                             var  orderindex : system_ihr_array_integer );
```

```
{ subroutine to order a list, least first }
{ input numorder = the number of items to be ordered  }
{ input ordertable = the items to be ordered }
{ output orderindex = pointer to order value table }
{ nxt = Table used in order subroutine }
```

```
var
  stop:boolean;
  i,j,top,last,idx:integer;
  nxt : system_ihr_array_integer;
```

begin

for i := 1 to numorder do begin

if (i <= 1) then begin

top := 1;

nxt[1] := 0;

end

else begin

j := top;

last := 0;

repeat

stop := true;

if (ordertable[i] > ordertable[j]) then begin

last := j;

j := nxt[j];

stop := (j = 0);

if (stop) then begin

nxt[last] := i;

nxt[i] := 0;

end

end

```

else begin
  if (j <> top) then begin
    nxt[ last ] := i;      { j not = top }
    nxt[ i ] := j;
  end
else begin
  top := i;                { j = top }
  nxt[ i ] := j;
end;
end;
until stop;
end;
end;
indx := 1;
j := top;
repeat
  orderindex[ indx ] := j;
  j := nxt[ j ];
  indx := indx + 1;
until (j = 0);
end;

procedure inverse_ihr_ftn( i : integer;
                           unitihr : real;
                           var unitmw : real );

{ Routine to return unit MW given unit incremental heat rate}
{ input : unit number = i}
{   unit inc heat rate = unitihr}
{ output: unit MW stored in unitmw}

label return;

var
  unitihr1, delihr, partihr, dihrdp : real;
  segmentihr : real;
  j, step : integer;

begin

```

```

if unitihr >= maxihr[ i ] then
begin
unitmw := pmax[ i ];
goto return
end;
if unitihr <= minihr[ i ] then
begin
unitmw := pmin[ i ];
goto return
end;

case curvetype of
poly :           {Polynomial curve}
begin
if curveorder <= 1 then
begin
if unitihr > coeff[ i,1 ] then unitmw:=pmax[i] else unitmw:=pmin[i];
goto return
end;

if curveorder = 2 then
begin
unitmw := ( unitihr - coeff[ i,1 ] ) / ( 2.0 * coeff[ i,2 ] );
goto return
end;

        { for curves of order >= 3 search for unitmw using Newtons
method }

unitmw := ( pmin[ i ] + pmax[ i ] ) / 2.0;
step := 0;
repeat
step := step + 1;

unitihr1 := 0;           {Calc unitihr at unitmw as unitihr1}
for j := curveorder downto 2 do
unitihr1 := ( unitihr1 + j * coeff[i,j] ) * unitmw;
unitihr1 := unitihr1 + coeff[i,1];

```

```

delihr := unitihr - unitihr1;
if abs( delihr ) < ihr_tolerance then goto return;

```

```

dihrdp := 0; {Calc curve second derivative}
for j := curveorder downto 3 do
    dihrdp := ( dihrdp + j*(j-1) * coeff[i,j] ) * unitmw;
dihrdp := dihrdp + 2.0 * coeff[i,2 ];
unitmw := unitmw + delihr/dihrdp;

until step > 20;

```

```

goto return
end;

```

procedure Dynamic_technique_dispatch(var lambda : real);

```

var
    i, n, lossiter : integer;
    lambdamin, lambdamax, lambda2, basekonst1, delambda : real;
    lambda2start, deltalambda, targetgen, totnewload, pnnew, pnnew1 :
real;
    unitihr, unitmw, totalgen, deltap, ptot, pnnewtot : real;
    konstanta1, konstanta2, lagrange, konstanta3, konstanta4, lambda1
: real;
    endloop : boolean;

```

```

begin
    writeln (ff,'Dynamic Formulation Technique');
    writeln (ff);

    konstanta1 := 0;
    konstanta2 := 0;
    for i:= 1 to ngen do
        begin
            konstanta1 := konstanta1 +
((coeff[i,1]*fuelcost[i])/(2*coeff[i,2]*fuelcost[i]));
            konstanta2 := konstanta2 + (1/(2*coeff [i,2]*fuelcost[i]));
        end;
    end;

```



```

if schedtype = totgen then totnewload := newload;
if schedtype = totload then totnewload := newload + mwlosses;
lagrange := ( totnewload + konstanta1)/konstanta2;

totalgen := 0;
for i:= 1 to ngen do
begin
unitmw := (lagrange-(coeff [i,1]*fuelcost[i]))/(2*coeff[i,2]*fuelcost[i]);
p[i] := unitmw ;
if p[i] <= pmin [i] then p[i]:= pmin[i];
if p[i] >= pmax [i] then p[i]:= pmax[i];
totalgen := totalgen + p[i] ;
end;
end; {end dynamic technique}

```

procedure base_point_dispatch(var lambda : real);

```

var
i, n, lossiter : integer;
lambdamin, lambdamax,deltap, lambda2,unitmw1 : real;
lambdastart, deltalambda, targetgen, delbase,delambda : real;
unitihr, unitmw, totalgen, lamresult, baseround : real;
basekonst1, deltabase, lagrange, totnewload : real;
endloop : boolean;
limittxt : string[5];
totalcost, totalload : real;
unitinccost, unitcost : real;

begin
writeln (ff,'Base Point and Participation Factor');
writeln(ff);

for i := 1 to ngen do          { Set unit output to midrange}
begin
p[ i ] := ( pmin[ i ] + pmax[ i ] ) / 2.0
end;

lossiter := 0;

```

```
endloop := false;
```

```
repeat {Top of iterative loop with losses}
```

```
lambdamin := 10000.0;
```

```
lambdamax := 0.0;
```

```
mwlosses := 0 ;
```

```
if losstype = lossform then { Calc losses and pen factors}
```

```
begin
```

```
loss_matrix_ftn;
```

```
writeln(ff);
```

```
writeln(ff,'mw losses = ',mwlosses:10:1);
```

```
end;
```

```
for i := 1 to ngen do {Calculate max and min lambdas}
```

```
begin
```

```
lambda := maxihr[ i ] * penfac[ i ] * fuelcost[ i ];
```

```
if lambda > lambdamax then lambdamax := lambda;
```

```
lambda := minihr[ i ] * penfac[ i ] * fuelcost[ i ];
```

```
if lambda < lambdamin then lambdamin := lambda
```

```
end;
```

```
writeln(ff,' lambda limits = ',lambdamin:10:4,lambdamax:10:4);
```

```
lambdastart := ( lambdamax + lambdamin ) / 2.0;
```

```
deltalambda := ( lambdamax - lambdamin ) / 2.0;
```

```
writeln(ff,' lambdastart deltalambda =
```

```
','lambdastart:10:4,deltalambda:10:4);
```

```
{Set up total generation target}
```

```
if schedtype = totgen then targetgen := schedmw;
```

```
if schedtype = totload then targetgen := schedmw + mwlosses;
```

```
{Lambda search}
```

```
lambda := lambdastart;
```

```
writeln(ff,' targetgen = ',targetgen:10:1);
```

```

n := 0;
repeat                                     {Top of lambda search loop}
    n := n + 1;
    totalgen := 0;
    for i := 1 to ngen do
        begin
            unitihr := lambda / ( penfac[ i ] * fuelcost[ i ] ) ;
            inverse_ihr_ftn( i, unitihr, unitmw ); {For given unitihr get
unitmw}
            p[ i ] := unitmw;
            totalgen := totalgen + p[ i ]
        end;

        writeln(ff,' lambda = ',lambda:10:4,' totalgen = ',totalgen:10:1);

    if abs( totalgen - targetgen ) >= total_gen_tolerance then
        begin
            if totalgen > targetgen then lambda := lambda - deltalambda;
            if totalgen < targetgen then lambda := lambda + deltalambda;
            deltalambda := deltalambda / 2.0
        end;

    until ( abs( totalgen - targetgen ) < total_gen_tolerance ) or
        ( n > 20 ) ;

    {See if another loss iteration is needed}

    if losstype <> lossform then endloop := true;
    lossiter := lossiter + 1;
    if lossiter > 10 then endloop := true

until endloop;

//writeln(ff,' lambda = ',lambda:10:4);
writeln( ff);
writeln(ff,
    'generator output limit inc cost penalty fact operating cost');

```

```

writeln(ff,
'      mw      $/mwhr      $/hr      ');
writeln(ff,
'-----');

totalgen := 0.0;
totalcost := 0.0;

for i := 1 to ngen do
begin
write(ff, genname[i]:9);
write(ff, ' ', p[i]:6:3, ' ');
limittxt := ' ';
if abs( p[i] - pmin[i] ) < total_gen_tolerance then limittxt := 'min ';
if abs( p[i] - pmax[i] ) < total_gen_tolerance then limittxt := 'max ';
write(ff, limittxt );

ihr_ftn( i, p[ i ], unitihr ); {Get unit incremental heat rate}

unitinccost := unitihr * fuelcost[i];
write(ff, unitinccost:9:4);
write(ff, ' ', penfac[i]:9:4, ' ');

prod_cost( i, p[ i ], unitcost ); {Calculate unit operating cost}

writeln(ff, "unitcost:9:12);
totalgen := totalgen + p[i];
totalcost := totalcost + unitcost;
end;
writeln(ff,
'-----');
write(ff, ' totals');
writeln(ff, totalgen:9:1,
' ', totalcost:9:12);
writeln(ff, ' lambda = ', lambda:10:4);
writeln(ff);
{end lambda}

```



```

basekonst1 := 0; {base point}
if schedtype = totgen then totnewload := newload;
if schedtype = totload then totnewload := newload + mwlosses;

for i:= 1 to ngen do
begin
if p[i] > pmax[i] then basekonst1 := basekonst1 + (1/(2*coeff
[i,2]*fuelcost[i]));
end;

delbase := totnewload - schedmw ;

for i:=1 to ngen do
begin
if p[i] > pmax[i] then
begin
lamresult := p[i]*100 ;
baseround := (round (lamresult) /100) ;
unitmw := baseround + (delbase/(2*coeff[i,2]*basekonst1*fuelcost[i]));
p[i] := unitmw ;
end;
totalgen := totalgen + p[i] ;
end;
end; {end base point}
procedure ihr_ftn(      i : integer;
                    unitmw : real;
                    var unitihr : real );

    { Routine to return unit incremental heat rate given unit output in
    MW }
    { input : unit index = i}
    {      unit mw = unitmw}
    { output: unit incremental heat rate = unitihr}

var
    partmw : real;
    j : integer;

begin

```

```

case curvetype of
  poly :           {Polynomial I/O curve}
    begin
      unitihr := 0.0;
      for j := curveorder downto 2 do
        unitihr := ( unitihr + j * coeff[ i,j ] ) * unitmw;
        unitihr := unitihr + coeff[ i,1 ]
      end;

      end; { End of case statement}
    end; { End ihr_ftn procedure }

procedure datadump( var outfile:text );
var
  i,j : integer;

begin
  writeln(outfile);
  writeln(outfile, title1);
  writeln(outfile, title2);
  writeln(outfile);
  writeln(outfile, ' number of generator units = ',ngen );

  case curvetype of
    poly : writeln(outfile, ' unit curve type = poly ');
    pinc : writeln(outfile, ' unit curve type = pinc ');
    pio  : writeln(outfile, ' unit curve type = pio');
    end; { End of case statement}

  writeln(outfile, ' curve order = ',curveorder);

  case losstype of
    noloss : writeln(outfile, ' network loss representation = noloss ');
    constpf : writeln(outfile, ' network loss representation = constpf ');
    lossform : writeln(outfile, ' network loss representation = lossform ');
    end; { End of case statement}

  for i := 1 to ngen do

```

```

begin
  writeln(outfile);
  write(outfile, genname[i], ' limits = ', pmin[i]:7:2, ' ', pmax[i]:7:2 );
  writeln(outfile, ' fuelcost = ', fuelcost[i]:10:4 );
  case curvetype of
    poly :
      begin
        writeln(outfile, ' polynomial coefficients' );
        for j := 0 to curveorder do
          begin
            writeln(outfile, coeff[i,j]:15:6);
          end;
        end;
      begin
        writeln(outfile, ' incremental cost curve points' );
        writeln(outfile, 'input at pmin = ', minput[i]:10:2);
        for j := 0 to curveorder do
          begin
            writeln(outfile, ihr_mwpoint[i,j]:9:2, ihr_cost[i,j]:9:3 )
          end;
        end;
        writeln(outfile);
      end;
    pio :
      begin
        writeln(outfile, ' cost curve points' );
        for j := 0 to curveorder do
          begin
            writeln(outfile, io_mwpoint[i,j]:9:2, ' ', io_cost[i,j]:9:3 )
          end;
        end;
        writeln(outfile);
      end;
    {end of case statement }
  end;
case losstype of
  constpf :
    begin
      writeln(outfile);
    end;
  end;

```

```

writeln(outfile,' Penalty Factors');
for i := 1 to ngen do
begin
  writeln(outfile,' penalty factor ',i,' ',penfac[i]:10:3)
end;
writeln(outfile);
end;

lossform :
begin
  writeln(outfile);
  writeln(outfile,' Loss Formula');
  writeln(outfile,'B00 = ',b00:10:4);
  writeln(outfile);
  writeln(outfile,'B0 = ');
  for i := 1 to ngen do
    if i < ngen then
      write(outfile,b0[i]:10:4,',')
    else
      writeln(outfile,b0[i]:10:4);
  writeln(outfile);
  writeln(outfile,' B = ');
  for i := 1 to ngen do
    begin
      for j := 1 to ngen do
        if j < ngen then
          write(outfile,b[i,j]:10:4,',')
        else
          writeln(outfile,b[i,ngen]:10:4);
      end;
      writeln(outfile)
    end;
  end; { End of case statement}
  // if solution_type = lamsearch then writeln(outfile,'using lambda
search')
  // else writeln(outfile,'using tablelookup');
  writeln(outfile);
  if solution_type = dynamictechnique then
    begin

```



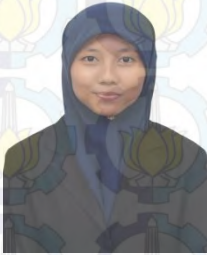
```

if schedtype = totgen then
    writeln(outfile, ' total generation schedule = ',newload:10:1)
else
    writeln(outfile, ' total load schedule = ', newload:10:1);
end
else begin
    if schedtype = totgen then
        writeln(outfile, ' total generation schedule = ',schedmw:10:1)
    else
        writeln(outfile, ' total load schedule = ', schedmw:10:1);
    end ;
    if losstype = lossform then writeln(outfile, ' using loss formula ')
        else writeln(outfile,' losses neglected');
    writeln(outfile);
end; { End procedure }

end.

```

RIWAYAT HIDUP PENULIS



OKTARINA RATRI WIJAYANTI, lahir di Surabaya, 20 Oktober 1992. Penulis tamat dari bangku sekolah dasar di SDN Latsari I Tuban pada tahun 2004 dan melanjutkan di sekolah menengah pertama di SMPN 3 Tuban, lulus tahun 2007. Setelah lulus SMP, penulis melanjutkan sekolah ke SMAN 1 Tuban. Setelah lulus pada tahun 2010, penulis melanjutkan studi D3 di Politeknik Elektronika Negeri Surabaya dan lulus pada tahun 2013, kemudian melanjutkan S1 di Institut Teknologi Sepuluh Nopember (ITS) Surabaya jurusan Teknik elektro dan mengambil konsentrasi dalam Bidang Studi Teknik Sistem Tenaga. Penulis dapat dihubungi melalui alamat email : oktarinaratri@gmail.com